Instituto Superior
de Engenharia
Politécnico de Coimbra

DEPARTAMENTO DE ENGENHARIA
ELETROTÉCNICA

Automatização de um Sistema de Mistura

Relatório de Estágio para a obtenção do grau de Mestre em Engenharia Eletrotécnica

Especialização em Automação e Comunicações em Sistemas Industriais

Autor

Leandro Miguel Jesus da Costa Dias

Orientador

Doutor Inácio de Sousa Adelino da Fonseca

Professor do Departamento DEE

Instituto Superior de Engenharia de Coimbra

Supervisor na empresa

Engenheiro Luís Fernandes

Tecnocon, S.A.

Coimbra, março, 2024

INSTITUTO POLITÉCNICO
DE COIMBRA

INSTITUTO SUPERIOR
DE ENGENHARIA
DE COIMBRA

Resumo

O presente relatório documenta o resultado do trabalho desenvolvido durante o estágio curricular realizado na empresa Tecnocon, S.A. no âmbito do Mestrado em Engenharia Eletrotécnica do ISEC.

Durante o estágio, o autor teve a oportunidade de aplicar e aprofundar os seus conhecimentos em automação industrial, particularmente na concepção, configuração e otimização de aplicações de consolas HMI através do WinCC. A experiência incluiu a colaboração em projetos inovadores, nos quais o autor foi responsável por projetar *interfaces* intuitivos e eficientes que facilitaram a supervisão e o controle de processos industriais complexos. O presente relatório ilustra o caso de um sistema de mistura.

O autor trabalhou ativamente na integração de sistemas de automação, utilizando consolas HMI para proporcionar aos operadores uma visão abrangente e precisa do status do processo. Ao colaborar com equipes multidisciplinares, foi possível desenvolver soluções adaptadas às necessidades específicas de cada ambiente industrial, garantindo a eficiência operacional e a segurança do processo.

Além disso, a busca contínua por conhecimento levou a explorar tecnologias emergentes, como a integração de sistemas baseados em Wonderware e a implementação de abordagens inovadoras para melhorar a eficácia das *interfaces* HMI e desenvolvimento de soluções avançadas que impulsionaram o produto.

Em termos de tecnologias foi programado um autómato da marca Siemens S7-1200 usando o TIA Portal V16, consolas Siemens para implementar a interação com o operador, usando o WinCC, e finalmente o software Wonderware para desenvolver aplicações de SCADA.

Em resumo, o estágio de mestrado proporcionou um ambiente propício para aprofundar os conhecimentos e competências na área de automação, bem como para colaborar com os profissionais da empresa.

Palavras-chave: Automação, Consolas HMI, WinCC, Wonderware, Tia Portal.

Abstract

This report documents the results of the work carried out during the internship at Tecnocom, S.A. as part of ISEC's Master's Degree in Electrical Engineering.

During the internship, the author had the opportunity to apply and deepen his knowledge of industrial automation, particularly in the design, configuration and optimization of HMI console applications using WinCC. The experience included collaboration on innovative projects in which the author was responsible for designing intuitive and efficient interfaces that facilitated the supervision and control of complex industrial processes. This report illustrates the case of a mixing system.

The author has worked actively on the integration of automation systems, using HMI consoles to provide operators with a comprehensive and accurate view of the process status. By collaborating with multidisciplinary teams, it was possible to develop solutions adapted to the specific needs of each industrial environment, guaranteeing operational efficiency and process safety.

In addition, the continuous search for knowledge led to the exploration of emerging technologies, such as the integration of Wonderware-based systems and the implementation of innovative approaches to improve the effectiveness of HMI interfaces and the development of advanced solutions that drove the product forward.

In terms of technology, a Siemens S7-1200 PLC was programmed using TIA Portal V16, Siemens consoles to implement operator interaction using WinCC, and finally Wonderware software to develop SCADA applications.

In summary, the master's internship provided an environment conducive to deepening my knowledge and skills in the field of automation, as well as collaborating with the company's professionals.

Keywords: Automation, HMI, WinCC, Wonderware, Tia Portal.

Epígrafe

O começo de todas as ciências é o espanto de as coisas serem o que são.
Aristóteles

Agradecimentos

Todo o trajeto percorrido não depende apenas do esforço e empenho de quem o faz mas também de quem apoia, crítica e incentiva. Quero, por isso, expressar a minha gratidão e reconhecimento:

Em primeiro lugar ao Professor Doutor Inácio Fonseca, meu orientador, pela competência, pelos conhecimentos transmitidos, pelas reflexões partilhadas, pelo apoio, disponibilidade, estímulo, crítica e cordialidade com que sempre me recebeu, sem os quais a concretização deste trabalho não seria possível.

Quero agradecer também ao meu supervisor na Tecnocon, Eng.º Luís Fernandes e aos meus colegas do departamento de programação por todos os conhecimentos transmitidos, disponibilidade e orientação dada que foram imprescindíveis para o sucesso deste estágio.

Aos meus pais pelo facto de estarem sempre presentes ao longo do meu percurso académico e pelo apoio que me dão.

E por fim, agradecer a todos os colegas que tive o privilégio de conhecer no meu percurso académico, pela paciência, cumplicidade, disponibilidade e motivação.

Índice

Índice de figuras

Lista de abreviaturas

EEPROM	Electricaly Erasable Programable Read Only Memory
EPROM	Erasable Programable Read Only Memory
IEEE	*Institute of Electrical and Electronics Engineers*
ISEC	Instituto Superior de Engenharia de Coimbra
RAM	Random Access Memory
SCADA	Supervisory Control and Data Acquisition
UI	User Interface

Lista de siglas e acrónimos

CPU	Central Processing Unit
FBD	Function Block Diagram
HMI	Human Machine Interface
IEC	*International Electrotechnical Commission*
IL	Instruction List
LD	Ladder Diagram
PID	Proporcional, Integral, Derivativo
PLC	Programmable Logic Controller
SFC	Sequential Function Chart
ST	Structured Text

Lista de símbolos

kN	Quilonewton
ε_{ax}	Extensão axial (%)

1 Introdução

O presente documento surge com o objetivo de apresentar o trabalho desenvolvido durante o estágio, a fim da obtenção do grau de Mestre em Engenharia Eletrotécnica com especialização em Automação e Comunicações em Sistemas Industriais.

1.1 Enquadramento

A automação industrial representa uma revolução na forma como as indústrias operam e produzem bens e serviços. Essa transformação é impulsionada por avanços tecnológicos, como a robótica, a inteligência artificial, a Internet das coisas (IoT) e sistemas de controle sofisticados. Essas tecnologias combinadas têm o poder de otimizar processos, melhorar a eficiência e qualidade, aumentar a segurança e reduzir custos na produção industrial. Visa substituir ou aprimorar tarefas repetitivas, perigosas e monótonas que, antes, eram desempenhadas por trabalhadores, por meio da aplicação de máquinas e sistemas inteligentes. Ela abrange diversas áreas, desde manufatura e montagem de produtos até controle de processos químicos e industriais, logística, embalagem, entre outras atividades.

O uso de sensores e sistemas de controle avançados permite que as máquinas reajam em tempo real às mudanças nas condições de operação, otimizando a produção e minimizando desperdícios. Além disso, a integração de sistemas e a coleta de dados em tempo real permitem uma visão holística do processo produtivo, possibilitando tomadas de decisão mais informadas e estratégicas.

Com a automação industrial, é possível aumentar a produtividade, reduzindo o tempo de produção e melhorando a qualidade dos produtos, resultando em ganhos competitivos significativos para as empresas. Além disso, a automação contribui para a segurança dos trabalhadores, pois tarefas perigosas podem ser delegadas a máquinas, minimizando os riscos de acidentes de trabalho.

No entanto, esta área também traz desafios, como a necessidade de treino adequado para os operadores de sistemas automatizados, adaptação dos processos produtivos existentes para acomodar a nova tecnologia e questões relacionadas com a privacidade e segurança de dados.

Nesta era de transformação digital e inovação contínua, a automação industrial é uma das principais alavancas para impulsionar a eficiência e a competitividade das indús-

trias. Aqueles que souberem abraçar esta mudança e integrar as tecnologias de forma inteligente estarão preparados para enfrentar os desafios do futuro e colher os benefícios de uma produção mais eficiente, sustentável e lucrativa.

1.2 Objetivos

O principal objetivo do estágio compreende a realização de um conjunto de tarefas relacionadas com um dos principais serviços prestados pela empresa Tecnocon atualmente, a automação industrial. Assim sendo, o objetivo proposto foi a realização da automatização de um sistema de mistura. A principal tarefa tem o foco na criação de uma solução de programação para autómato com a elaboração de "*Human Machine Interfaces*" e Scada.

1.3 Empresa de Acolhimento

1.3.1 Caracterização

A TECNOCON [3] é uma empresa criada em 1989, onde a competência e a experiência dos seus técnicos, adquirida ao longo dos anos, tem sido um dos fatores de sucesso para um crescimento sustentado. Tem como atividade principal a automação industrial, que inclui o desenvolvimento e produção de soluções para automação e controlo de processos industriais, projeto e montagem de infraestruturas elétricas, assistência técnica e comercialização de componentes industriais.

1.3.2 Localização do estágio

O presente estágio foi realizado nas instalações da TECNOCON[1] a qual se encontra sediada no parque industrial de Vale de Cambra, em Aveiro, sendo a sua localização representada na Figura 1.1.

[1] https://www.tecnocon.pt/?l=en

Figura 1.1: Vista aérea do parque industrial de Codal, Vale de Cambra e a empresa Tecnocon

1.4 Estrutura do relatório

O relatório encontra-se organizado em cinco capítulos.

O capítulo um apresenta um enquadramento geral ao tema e os objetivos a alcançar com a sua realização.

O capítulo dois apresenta algumas tecnologias da área de automação nomeadamente as Linguagens de Programação de autómatos.

O capítulo três foca as tecnologias utilizadas, desde o S7-1200 da Siemens, WinCC e o *Wonderware System Platform* [4].

O capítulo quatro documenta o desenvolvimento efetuado para o sistema de mistura.

O capítulo cinco apresenta as conclusões.

2 Estado da Arte e conceitos

O presente capítulo descreve as ferramentas e o estado recente das tecnologias usadas no âmbito da automação industrial.

2.1 História

A ideia de criar sistemas flexíveis capazes de controlar processos, [5] sempre norteou o espírito humano. Foi nos sistemas mecânicos que primeiro se desenvolveu este princípio.

À medida em que se foi conhecendo e desenvolvendo toda a tecnologia dos circuitos elétricos, logo se verificou que facilmente se poderia alterar as características de um circuito de controlo, recorrendo a relés e comutadores. Desta forma, diversas combinações nos comutadores, davam origem a diferentes modos de funcionamento. Era o primeiro indício de "programação".

De acordo com as necessidades de controlo, foram-se desenvolvendo componentes tais como temporizadores, contadores, relés biestáveis e um sem número de outros componentes que iam integrando sistemas cada vez mais complexos.

Ainda hoje podemos admirar em alguns equipamentos "programadores de pinos". Estes programadores muito rudimentares, constavam de uma matriz de condutores que eram intercetados por uma cavilha ou pino que de acordo com a sua posição permitia definir a ativação de determinado circuito.

Em paralelo ao desenvolvimento de circuitos elétricos, apareceram também circuitos pneumáticos. Tal como nos circuitos elétricos, nestes também foram desenvolvidos uma série de elementos que tinham desempenhos idênticos aos componentes elétricos.

A invenção da Unidade Central de Processamento (1938), [6] dá por sua vez origem ao microprocessador. É com base neste componente que em 1969-70 aparecem nos EUA os primeiros autómatos programáveis. Foi a indústria automóvel quem primeiro os utilizou. Na Europa, só dois anos depois é que começam a ser empregues na indústria.

Hoje, já é difícil falar em automatização industrial sem que se tenha de referir o autómato programável.

Contrastando com a tecnologia cablada, o autómato programável tem inúmeras vantagens:

- É muito mais fiável, pois o número de componentes mecânicos e de ligações é mínimo.
- O desenvolvimento do programa pode ser feito em paralelo com a montagem dos equipamentos. O sistema por lógica cablada, só pode ser montado depois do projeto estar completamente concluído.
- As alterações do automatismo só implicam alterações no programa. Na lógica cablada, qualquer alteração implica a alteração da cablagem e dos componentes.
- O espaço ocupado pelo autómato é constante e independente da complexidade da lógica do automatismo.
- Não requer stocks de equipamento de reserva tão elevados como nos sistemas por lógica cablada.

Sobre os sistemas controlados por microprocessador ou microcomputador, o autómato não requer a presença de um perito em informática ou em assembler para programar ou alterar um programa. A linguagem utilizada é standard e pode ser facilmente apreendida por pessoas com uma formação mínima.

Pode classificar-se a utilização dos autómatos programáveis em três classes:

- Controlo de máquinas ou automatismos simples e individualizados.
- Controlo e sincronização de diversas máquinas ou automatismos de uma linha de fabrico ou de um sistema complexo.
- Supervisão e controlo de uma unidade fabrico ou de um conjunto de sub-sistemas que podem eles mesmos serem também controlados por autómatos.

2.2 Linguagens de programação

Em termos de linguagens de programação a norma IEC 61131-3 [7] descreve a sintaxe das várias linguagens que se apresentam nas sub-seções seguintes. Este tema será aprofundado no Capítulo 3.

2.2.1 Ladder Logic

O Ladder [7] é a linguagem de programação mais tradicional e amplamente utilizada em *Programmable Logic Controllers* (PLCs). Esta linguagem assemelha-se a um diagrama de circuitos elétricos em forma de "escada", com contatos, bobinas e relés representados por símbolos gráficos. É especialmente adequada para aplicações que envolvem lógica booleana, como portas lógicas e temporizadores.

2.2.2 Instruction List

A linguagem de programação IL [7] é baseada em texto e utiliza uma lista de instruções mnemónicas para descrever as operações do PLC. É uma linguagem mais próxima da linguagem de baixo nível, o que permite maior controle sobre os detalhes do hardware do PLC.

2.2.3 Structured Text

O ST [7] é uma linguagem de programação que se assemelha a linguagens de programação de alto nível, como C ou Pascal. Permite a escrita de algoritmos complexos e é mais adequada para aplicações que requerem cálculos e manipulação de dados.

2.2.4 Function Block Diagram

A linguagem FBD [7] é baseada em blocos de funções interconectados, cada um representando uma função específica. É uma abordagem gráfica que permite criar diagramas mais complexos e modulares.

2.2.5 Sequential Function Chart

O SFC [7] é uma linguagem que permite descrever o comportamento do PLC em termos de estados e transições entre esses estados. É particularmente útil para sistemas que envolvem sequências de operações ou máquinas com ciclos predefinidos.

2.3 Revisão do estado atual para HMI e SCADA

Na área da automação, existem diversas ferramentas interessantes para o desenvolvimento de aplicações *Human Machine Interface*. Destacam-se as seguintes:

- WinCC (Siemens) [8]: é uma das ferramentas usada neste trabalho, que permite desenvolver aplicações para consolas da Siemens e ainda, desenvolver aplicações SCADA (software executado num desktop). Esta aplicação permite o desenvolvimento do UI através da introdução dos elementos gráficos, permite a apresentação de dados recolhidos nos equipamentos, entre outras funcionalidades que sejam programadas.
- Wonderware [4]: O Wonderware fornece uma variedade de soluções de software para automação industrial, incluindo HMI/SCADA, nomeadamente o InTouch.
- InTouch [9]: é um software HMI/SCADA que permite o desenvolvimento de aplicações de visualização de processo, controle e monitorização.

- Citect SCADA [10]: Citect SCADA é uma solução da Schneider Electric que permite o desenvolvimento de aplicações de visualização de processo, controle e monitorização.
- VTScada [11]: é um software de supervisão que fornece uma solução HMI/SCADA sendo versátil e personalizável.
- FactoryTalk View [12]: O Factory Talk (Rockwell Automation) é um produto competidor do Wonderware, pois fornece várias ferramentas similares, sendo uma delas o FactoryTalk View. Este software permite o desenvolvimento de aplicações de SCADA.

2.4 Conclusão

Neste capítulo apresentou-se de forma sucinta algumas das tecnologias de desenvolvimento SCADA, os tipos de linguagens de programação de autómatos e um resumo histórico.

3 Tecnologias Utilizadas

No presente capítulo irão ser apresentadas as tecnologias utilizadas durante o trabalho realizado no período do estágio. Essencialmente, o capítulo estará focado em apresentar o Tia Portal, WinCC e as ferramentas do universo Wonderware.

3.1 Controladores Lógicos Programáveis

Figura 3.1: PLC S71200 [1]

A história dos controladores lógicos programáveis (PLCs) remonta ao final da década de 1960, quando na indústria de manufactura surgiu a necessidade de sistemas de controle flexíveis e reprogramáveis, pois os sistemas de controle em fábricas geralmente dependiam da lógica de relés. Esses sistemas eram compostos por relés e circuitos eletromecânicos, que eram difíceis de modificar e propensos a erros e uma reparação que era extremamente cara e demorada. A Figura 3.1 ilustra um PLC da Siemens.

Um PLC é um dispositivo computadorizado especializado usado em automação industrial para controlar e monitorizar máquinas e processos. É um computador digital que é projetado para operar em ambientes industriais agressivos e executar funções de controle em tempo real. O principal objetivo de um PLC é automatizar e agilizar processos industriais, recebendo entradas de sensores ou outros dispositivos, processando-os usando uma lógica programada e gerando saídas para controlar atuadores, motores e

outros equipamentos. Os PLCs são amplamente utilizados em várias indústrias, incluindo manufatura, energia, petróleo e gás, transporte e muito mais.

Praticamente, qualquer linha de produção, máquina ou processo podem ser grandemente melhorados pela utilização de PLCs. Dessa forma, entre os benefícios de se utilizar um PLC, estão a capacidade de reprogramação, alteração de sequências, ampliação de linhas, criação de réplicas de máquinas e processos, tudo isso enquanto podemos recolher e comunicar informações vitais.

3.1.1 Arquitetura

Um autómato programável é constituído basicamente por (Figura 3.2):

- Unidade central de processamento também denominada de CPU;
- Dispositivo de programação;
- Portas de Comunicação;
- Memória de programa e de dados;
- Entradas e saídas (I/O);
- Fonte de alimentação;
- Porta de periféricos.

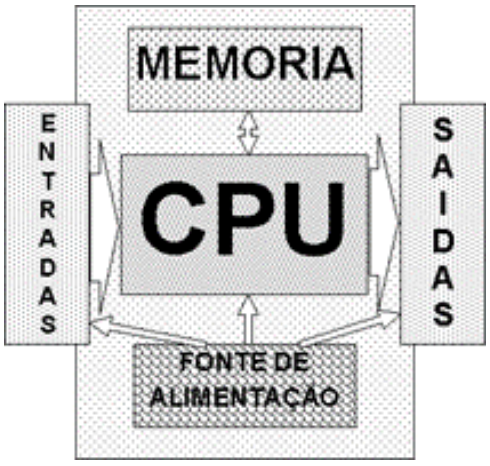


Figura 3.2: Estrutura de um autómato

Os PLC podem apresentar aspectos físicos diferentes, diferentes performances e custos muito díspares. No entanto, os seus elementos constituintes são fundamentalmente os mesmos.

Sendo um equipamento capaz de controlar processos, naturalmente dispõe de dispositivos de aquisição e saída de informações e sendo também um equipamento programável, integra um microprocessador e uma memória para guardar o programa. Para alimentar os circuitos atrás descritos, existirá também uma fonte de alimentação. Para

que possa ser introduzido o programa e para que possa existir um diálogo básico para o exterior, dispõe também da possibilidade de ligar dispositivos de programação.

CPU

Este bloco tem a função de ler os valores lógicos presentes nas entradas, executar as instruções que constituem o programa e transferir para as saídas as ordens provenientes dessas instruções. Tem ainda a seu cargo gerir todos os periféricos e diagnosticar defeitos que possam ocorrer internamente.

Na base de tudo isto, está um ou mais microprocessador que de uma forma sequencial vão executando instruções a velocidades extremamente elevadas.

A sequência descrita no primeiro parágrafo, é continuamente realizada. O tempo gasto para a realizar é designado como tempo de ciclo (ou tempo de scan) e é da ordem dos milisegundos.

O tempo de ciclo depende de muitos factores dos quais se destacam:

- Velocidade de trabalho do microprocessador(s);
- Número de instruções do programa;
- Tipo de instruções usadas no programa;
- Número de periféricos.

O CPU tem ainda circuitos de endereçamento de entradas e saídas, uma memória com o sistema operativo, interfaces para unidades externas, um circuito para a detecção de falhas de alimentação, e outros que interligam os anteriores.

Para sinalizar o estado de funcionamento do CPU, normalmente existem no frontal do mesmo, sinalizadores luminosos (led's).

É comum encontrar as seguintes sinalizações:

- Presença de alimentação (POWER);
- Erro no CPU (ERROR ou FAILURE);
- Execução do programa (RUN);
- Falha da bateria de backup (BATTERY).

Como o CPU é um elemento vital de um autómato, existem modelos que para garantirem uma grande fiabilidade de operação, dispõem de duplo CPU; quando um deles falha, o outro entra imediatamente em serviço, sem interromper o controlo do sistema.

Memória

É na memória que se encontra o programa a ser executado pelo autómato. A memória tem como função salvaguardar todas as instruções do programa, mesmo quando este

não está a ser alimentado.

A memória caracteriza-se pela sua capacidade que pode ser expressa de três formas:

- Número de bits ou Kbts (1 Kbts = 1024 bits)
- Número de Bytes ou KB (1 Byte = 8 bits)
- Número de Words ou KW (1 Word = 16 bits)

Quanto à sua tecnologia podem ser :

- RAM (Random Access Memory): Estas memórias têm a vantagem de poderem ser escritas e alteradas facilmente. São as mais usadas quando se está na fase de desenvolvimento do programa ou quando o sistema a controlar sofre frequentes alterações. Estas memórias perdem a informação quando a alimentação eléctrica das mesmas falha; por isso, obrigam ao uso de uma pilha de recurso que assegura a sua alimentação no caso de uma falha de energia.
- EPROM (Erasable Programable Read Only Memory): Esta memória não perde a informação nela gravada no caso de falhar a tensão. Têm como desvantagem o facto de ser muito morosa qualquer alteração, mesmo sendo de um só bit. Antes de ser programada por um equipamento próprio, tem de ser apagada por exposição aos raios ultravioletas.
- EEPROM (Electricaly Erasable Programable Read Only Memory): Esta memória não perde informação por falta de tensão de alimentação e pode ser apagada e escrita pelo autómato. Tem vantagens sobre os modelos anteriores, mas os inconvenientes de ter um número limitado de ciclos de escrita e do seu custo ser mais elevado que o de uma RAM.
- FLASHRAM: Esta memória de tecnologia muito recente, tem características semelhantes às EEPROM, permitindo também escrita e leitura no próprio circuito onde é usada. Limitada também pelo número de ciclos de escrita, apresenta vantagens sobre a EEPROM (uma delas, a velocidade de escrita).

A memória de um PLC desempenha um papel crucial no funcionamento eficaz do sistema de automação. Uma gestão adequada da memória é essencial para garantir que o programa de controle seja executado de forma consistente e eficiente. Também é importante dimensionar a memória corretamente de acordo com a complexidade do programa e as necessidades do sistema. O espaço de memória disponível pode afetar o número de variáveis, a complexidade da lógica, a quantidade de documentação e outros aspectos do projeto.

Entradas e saídas (I/O)

As entradas e saídas permitem que o PLC interaja com o ambiente externo, recebendo sinais dos sensores (entradas) e controlando dispositivos como motores e válvulas (saí-

das). Os módulos de E/S são fundamentais para monitorizar e controlar processos industriais de maneira eficaz. Os principais aspetos dos módulos de Entrada e Saída:

- Módulos de Entrada (E): Os módulos de entrada permitem que o PLC receba sinais do mundo externo, como sensores, interruptores, botões e outros dispositivos. As informações recolhidas pelas entradas são usadas pelo PLC para tomar decisões lógicas e controlar as saídas de acordo com a programação definida.
- Comunicação com o PLC: Os módulos de E/S comunicam com o PLC através de barramentos de comunicação internos. Eles enviam informações das entradas para a CPU do PLC, onde são processadas. A CPU, por sua vez, envia comandos para as saídas conforme a lógica programada.
- Configuração e Expansão: Os módulos de E/S podem ser configurados de acordo com as necessidades da aplicação. Eles podem ser adicionados ou removidos conforme necessário. Isso permite que os sistemas de automação sejam flexíveis e escalonáveis para atender a diferentes requisitos de controle.
- Módulos de Saída (S): Os módulos de saída permitem que o PLC controle dispositivos externos, como motores, solenoides, atuadores e válvulas. As saídas ativam ou desativam dispositivos conforme as decisões lógicas do programa.

3.1.2 Norma IEC 61131-3

A norma IEC 61131-3 [7] é um padrão internacional estabelecido pela Comissão Eletrotécnica Internacional (*International Electrotechnical Commission - IEC*) que define um conjunto de linguagens de programação padronizadas para programação de Controladores Lógicos Programáveis (PLCs) e sistemas de automação industrial.

Essa norma foi publicada pela primeira vez em 1993 e tem sido amplamente adotada na indústria para promover a interoperabilidade e a portabilidade de programas entre diferentes fabricantes de PLCs. O objetivo principal da IEC 61131-3 é oferecer uma estrutura comum para programação de PLCs, permitindo que os programadores desenvolvam lógica de controle usando uma variedade de linguagens padronizadas.

A IEC 61131-3 define cinco linguagens de programação que podem ser usadas para desenvolver programas de controle:

- Ladder Diagram (LD): É uma linguagem gráfica baseada em diagramas de escada, que se assemelham aos circuitos elétricos. Esta linguagem é amplamente usada e popular, especialmente em aplicações relacionadas com a automação industrial [7].
- Structured Text (ST): É uma linguagem de programação textual que se assemelha a linguagens de programação convencionais, como C ou Pascal. Esta linguagem oferece maior flexibilidade e expressividade em comparação com outras lingua-

gens [7].

- Function Block Diagram (FBD): É uma linguagem gráfica que permite aos programadores criar programas usando blocos funcionais interconectados, representando funções lógicas ou matemáticas [7].
- Instruction List (IL): É uma linguagem de programação textual que usa uma lista de instruções para descrever a lógica de controle. É semelhante à linguagem de baixo nível Assembly [7].
- Sequential Function Chart (SFC): É uma linguagem gráfica que descreve a lógica de controle em termos de estados, transições e ações associadas. É útil para descrever sequências complexas de operações [7].

A norma IEC 61131-3 também define requisitos para o ambiente de desenvolvimento de software, incluindo recursos como depuração, documentação e gestão de projeto. Ao seguir a IEC 61131-3, os fabricantes de PLCs podem garantir que seus dispositivos sejam compatíveis com uma ampla gama de ferramentas de programação e que os programas desenvolvidos numa determinada linguagem possam ser facilmente migrados para outros PLCs compatíveis. Em resumo, a norma IEC 61131-3 estabelece um conjunto de linguagens de programação padronizadas para programação de PLCs e sistemas de automação industrial, promovendo a interoperabilidade e a portabilidade de programas entre diferentes fabricantes. Isso facilita o desenvolvimento de lógica de controle e permite uma maior flexibilidade e eficiência na automação industrial.

3.2 Totally Integrated Automation Portal

Totally Integrated Automation Portal (TIA Portal) [13], [14] é uma plataforma de software desenvolvida pela Siemens para programação, configuração e diagnóstico de seus dispositivos de automação e controle. É uma estrutura de engenharia abrangente que fornece um ambiente unificado para projetar e comissionar soluções de automação.

O TIA Portal integra várias ferramentas de software que anteriormente estavam separadas, como o STEP 7 (para programação PLC), WinCC (para design HMI) e outros softwares de engenharia. Ao combinar essas ferramentas numa única plataforma, o TIA Portal simplifica o processo de engenharia, melhora a produtividade e permite a integração perfeita entre diferentes componentes de um sistema de automação. As ferramentas são:

- Programação PLC: O TIA Portal inclui o software de programação STEP 7, que permite aos utilizadores programar PLCs Siemens. Ele suporta várias linguagens de programação, incluindo lógica Ladder (LAD), diagrama de blocos de função (FBD), texto estruturado (ST) e gráfico S7.
- Design de HMI: Com o software WinCC integrado no TIA Portal, os utilizadores

podem projetar HMIs intuitivas e interativas para os sistemas de automação. As ferramentas de design HMI oferecem uma ampla gama de recursos, como gráficos, alarmes, tendências e gestão de receitas.

- Controle de movimento: O TIA Portal fornece recursos de controle de movimento para programação e configuração de servo-drives, motores e outros dispositivos de movimento. O TIA Portal permite que os utilizadores definam eixos de movimento, criem perfis de movimento e sincronizem vários eixos para controle preciso.
- Engenharia de segurança: O TIA Portal inclui ferramentas de engenharia de segurança para configurar e programar funções relacionadas com a segurança. Suporta o conceito de Segurança Integrada, permitindo aos utilizadores implementar funções de segurança em conformidade com as normas de segurança relevantes.
- Diagnóstico e manutenção do sistema: O TIA Portal oferece funções de diagnóstico abrangentes para monitorização e solução de problemas de sistemas de automação. Ele fornece diagnóstico do sistema em tempo real, gestão de alarmes e análise detalhada de erros para ajudar na manutenção e na deteção de falhas.
- Simulação e Comissionamento Virtual: O TIA Portal permite que os utilizadores simulem e validem os seus projetos de automação antes da implementação em ambiente real. Inclui recursos de comissionamento virtual, permitindo que os utilizadores testem o comportamento e o desempenho dos sistemas sem hardware físico.

3.3 WinCC

WinCC [15] é um pacote de software desenvolvido pela Siemens como parte do seu Portal de Automação Totalmente Integrada (TIA). É projetado especificamente para a criação de interfaces homem-máquina (HMIs) para sistemas de automação industrial. O WinCC permite que os utilizadores projetem, configurem e operem interfaces visuais que permitem que os operadores interajam e monitorizem processos industriais.[16]

- HMI Design: WinCC fornece um conjunto abrangente de ferramentas para criar HMIs visualmente atraentes e fáceis de usar. Ele oferece uma ampla gama de elementos gráficos, incluindo botões, interruptores, gráficos e tendências, para projetar interfaces intuitivas.
- Visualização de dados: WinCC permite a visualização de dados em tempo real de sistemas de automação. Ele permite a exibição de variáveis de processo, alarmes, informações de status e tendências históricas para fornecer aos operadores uma visão geral clara dos processos industriais.

- Gestão de alarme: WinCC inclui recursos avançados de gestão de alarme. Ele permite a configuração de limites de alarme, prioridades e configurações de notificação. Os operadores podem facilmente reconhecer e responder a alarmes, garantindo uma reação rápida a situações anormais.
- Registo de dados e relatórios: o WinCC permite a salvaguarda de dados de processo para fins de análise e relatórios. Ele pode armazenar dados históricos em base de dados e gerar relatórios, ajudando a identificar tendências, analisar o desempenho e tomar decisões informadas.
- Monitorização e Controlo Remotos: o WinCC suporta monitorização e controlo remotos de processos industriais. Permite que os utilizadores acedam e interajam com HMIs de locais remotos, permitindo a solução de problemas e operação remota.
- Segurança e Controle de Acesso: WinCC fornece recursos de segurança para proteger a HMI e os sistemas de automação subjacentes. Inclui funcionalidades de gestão de utilizadores, mecanismos de autenticação e definições de controlo de acesso para garantir a segurança adequada do sistema.

O WinCC é amplamente utilizado em indústrias como fabricação, energia, transporte e automação de processos. Este programa integra-se perfeitamente com outros componentes do TIA Portal, permitindo um ambiente de engenharia unificado para sistemas de automação.

3.4 Wonderware System Platform

Wonderware System Platform [4] é uma plataforma de software desenvolvida pela Aveva, fornecedora líder de soluções de automação industrial e gestão da informação. É projetado para criar e gerir aplicações de automação industrial e sistemas de controle de supervisão e aquisição de dados (SCADA).

Fornece um ambiente unificado e escalável para desenvolver, implantar e gerenciar aplicativos em vários setores e indústrias. Ele oferece uma ampla gama de recursos e capacidades para apoiar o projeto, visualização, controle e análise de processos industriais. As principais características são:

- Desenvolvimento unificado de aplicações: Wonderware System Platform oferece um ambiente de desenvolvimento unificado que permite aos utilizadores criar soluções usando ferramentas e métodos padronizados. Ele simplifica o processo de desenvolvimento, fornecendo uma interface consistente e modelo de programação.
- Escalabilidade e modularidade: A plataforma foi projetada para ser escalável e modular, permitindo que os utilizadores expandam e adaptem facilmente as suas

aplicações às novas necessidades. A ferramenta suporta arquiteturas distribuídas e pode lidar com programas que vão desde pequenos sistemas autónomos até grandes implementações em toda a empresa.

- Integração de dados e conectividade: A Wonderware System Platform fornece amplos recursos de integração de dados, permitindo conectividade perfeita a uma ampla gama de dispositivos industriais, base de dados e sistemas de terceiros. Ele suporta padrões abertos, como OPC, ODBC e serviços da Web para troca de dados.
- Visualização e HMI: A plataforma inclui poderosas capacidades de visualização e HMI (Human-Machine Interface). Ele permite que os utilizadores criem exibições gráficas intuitivas, painéis interativos e visualizações em tempo real de processos industriais. Ele também suporta gestão de alarme e notificação de eventos.
- Historiador e Análise de Dados: A Wonderware System Platform inclui um historiador de dados para armazenar e analisar dados de processos. Ele permite que os utilizadores capturem, armazenem e recuperem dados históricos para relatórios, análises e otimização de processos industriais.
- Integração com Sistemas MES e ERP: A plataforma suporta a integração com sistemas de Manufacturing Execution Systems (MES) e Enterprise Resource Planning (ERP), permitindo a troca de dados e o fluxo de informações entre diferentes níveis da organização.
- Segurança e conformidade: A Wonderware System Platform incorpora recursos de segurança robustos para proteger dados críticos e garantir a conformidade com as regulamentações do setor. Inclui autenticação de utilizadores, controlo de acesso e mecanismos de encriptação de dados para salvaguardar processos industriais.

O Wonderware System Platform é amplamente utilizado em indústrias como manufatura, energia, água e águas residuais, petróleo e gás, e muito mais. Ele fornece uma solução abrangente para automação e controle industrial, permitindo que as organizações melhorem a eficiência operacional, otimizem processos e alcancem melhores capacidades de tomada de decisão.

3.5 Wonderware Historian

O Wonderware Historian é um produto de software especializado, desenvolvido pela Aveva, a mesma empresa que oferece o Wonderware System Platform. É um historiador de dados de alto desempenho projetado para adquirir, armazenar e recuperar grandes volumes de dados de séries cronológicas gerados por processos e equipamentos industriais.

O principal objetivo do Wonderware Historian é fornecer uma maneira confiável e eficiente de capturar e arquivar dados de processo ao longo do tempo. Esses dados podem incluir vários tipos de informações, como temperatura, pressão, vazões, quantidades de produção e outras variáveis de processo, coletadas em intervalos regulares. O Historian contempla:

- Recolha de dados de alta velocidade: o Historian é otimizado para salvaguardar grandes quantidades de dados em altas velocidades, mesmo de processos industriais acelerados. Ele pode capturar dados em intervalos de milissegundos, permitindo a captura de alterações críticas do processo.

- Compressão de dados: O historiador usa técnicas avançadas de compressão de dados para armazenar com eficiência grandes volumes de dados. Ele pode reduzir significativamente os requisitos de armazenamento, preservando a fidelidade dos dados originais.

- Arquivamento e recuperação de dados: o historiador armazena dados em um formato de base de dados de séries temporais, facilitando a recuperação de dados históricos para fins de análise e relatórios. Os utilizadores podem acessar e recuperar dados de diferentes intervalos de tempo e usá-los para análise de tendências e otimização de desempenho.

- Contexto dos dados: O historian permite a associação de informações contextuais com dados, facilitando a compreensão do contexto dos eventos registados. Essas informações contextuais podem incluir detalhes sobre equipamentos, condições do processo, informações de lote e muito mais.

- Integração com outros sistemas: O Wonderware Historian integra-se perfeitamente com outros produtos Aveva e Wonderware, bem como com aplicações de terceiros. Ele pode conectar-se a sistemas SCADA, aplicativos HMI, sistemas de execução de fabricação (MES), sistemas de planejamento de recursos empresariais (ERP).

- Escalabilidade: o Historian é escalável, permitindo que as organizações o implantem em aplicativos pequenos de um único local ou em grandes arquiteturas distribuídas. Ele pode lidar com dados de um único dispositivo ou coletar dados de vários sites.

- Segurança de dados: O historian inclui recursos de segurança para proteger a integridade e a confidencialidade dos dados armazenados. Fornece controle de acesso, criptografia de dados e trilhas de auditoria para garantir a conformidade com as regulamentações do setor e os padrões de segurança cibernética.

3.6 Wonderware InTouch HMI

Wonderware InTouch HMI (*Human-Machine Interface*) é um pacote de software desenvolvido pela Aveva, anteriormente Wonderware, que faz parte de soluções de automação industrial e gestão de informações. O InTouch foi projetado para criar interfaces visuais poderosas e intuitivas que permitem que os operadores interajam e monitorizem processos industriais e máquinas. Este pacote contempla:

- HMI Design: InTouch fornece um ambiente amigável e flexível para projetar interfaces gráficas interativas. Ele permite que os utilizadores criem exibições dinâmicas com elementos gráficos, como botões, interruptores, tendências, gráficos e animações.
- Visualização de dados em tempo real: o InTouch permite a visualização de dados em tempo real, exibindo variáveis críticas de processo, alarmes e informações de status de forma clara e fácil de entender. Os operadores podem monitorizar o estado atual do processo industrial e responder a quaisquer anomalias ou alarmes prontamente.
- Gestão de alarmes: O InTouch inclui capacidades avançadas de gestão de alarmes. Ele permite que os utilizadores configurem limites de alarme, prioridades e notificações. Os operadores podem reconhecer e responder rapidamente a alarmes, garantindo uma ação atempada para situações anormais.
- Dados históricos e de tendências: o InTouch fornece recursos de tendências e dados históricos, permitindo que os operadores visualizem tendências históricas e analisem dados de processo ao longo do tempo. Isso ajuda a identificar padrões, anomalias e oportunidades de melhoria de processos.
- Scripts e lógica: o InTouch suporta scripts e operações lógicas, permitindo que os utilizadores criem comportamentos personalizados e automatizem determinadas ações com base em condições ou eventos específicos.
- Comunicação e conectividade de dados: O InTouch pode comunicar com uma ampla gama de dispositivos e sistemas industriais, incluindo PLCs, DCS (Distributed Control Systems) e sistemas SCADA. Ele suporta vários protocolos de comunicação para garantir a troca de dados sem interrupções.
- Segurança e gestão de utilizadores: o InTouch inclui recursos de segurança para controlar o acesso do utilizador e proteger dados confidenciais. Ele permite que os administradores efetuem a gestão de contas de utilizador e definam níveis de acesso para diferentes operadores.
- Suporte multilíngua: O InTouch oferece suporte multilíngua, permitindo que os utilizadores criem aplicativos HMI em diferentes idiomas para acomodar operações globais.

3.7 Microsoft SQL Server Management Studio

O Microsoft SQL Server Management Studio (SSMS) é um aplicativo de software desenvolvido pela Microsoft que fornece uma interface gráfica do utilizador (GUI) para gerir e administrar base de dados do Microsoft SQL Server. É uma ferramenta poderosa usada por administradores de base de dados, programadores e analistas para interagir com base de dados SQL Server e executar várias tarefas relacionadas à gestão e desenvolvimento de base de dados. Simplifica as tarefas de administração de base de dados, simplifica as atividades de desenvolvimento e fornece um conjunto abrangente de ferramentas para gerir e otimizar base de dados do SQL Server.

- Gestão de base de dados: o SSMS permite que os utilizadores criem, modifiquem e excluam base de dados, tabelas, exibições, procedimentos armazenados e outros objetos de base de dados. Ele fornece um vasto conjunto de ferramentas para design e gestão de esquemas de base de dados.
- Editor de consultas: o SSMS inclui um editor de consultas que permite aos utilizadores escrever e executar consultas Transact-SQL (T-SQL). Ele fornece realce de sintaxe, trechos de código e planos de execução de consultas para ajudar a otimizar o desempenho da consulta.
- Backup e recuperação de base de dados: o SSMS permite que os utilizadores executem backups e recuperações de base de dados, habilitando recursos de proteção de dados e recuperação de desastres.
- Gestão de segurança: com o SSMS, os administradores podem gerir contas de utilizador, funções e permissões para controlar o acesso à base de dados e garantir a segurança dos dados.
- Monitorização de desempenho: o SSMS fornece recursos de monitorização e ajuste de desempenho para ajudar a identificar e resolver limitações de desempenho em base de dados do SQL Server.
- Importar e exportar dados: o SSMS permite que os utilizadores importem e exportem dados entre base de dados do SQL Server e várias outras fontes de dados, facilitando a migração de dados e as tarefas de integração.
- Relatórios e análises: o SSMS inclui funcionalidades do Reporting Services (SSRS) e do Analysis Services (SSAS), permitindo que os utilizadores criem e gerenciem relatórios e projetos de análise de dados.
- Integration Services: o SSMS integra-se ao SQL Server Integration Services (SSIS), fornecendo ferramentas para projetar, implantar e gerir pacotes ETL (Extrair, Transformar, Carregar).

3.8 Conclusão

Este capítulo apresentou as tecnologias essenciais ao desenvolvimento do trabalho apresentado no documento.

4 Projeto desenvolvido

O presente capítulo documenta uma aplicação de automação no âmbito de um sistema de mistura. Essencialmente, o capítulo estará focado em detalhar aspetos técnicos, como apresentar receitas nas consolas, a interação com o autómato e com o sistema SCADA usando para isso variáveis denominadas *Tags* e protocolos de comunicação industriais, que no presente caso é o ProfiNet da Siemens, face aos equipamentos envolvidos.

4.1 Sistema de Mistura

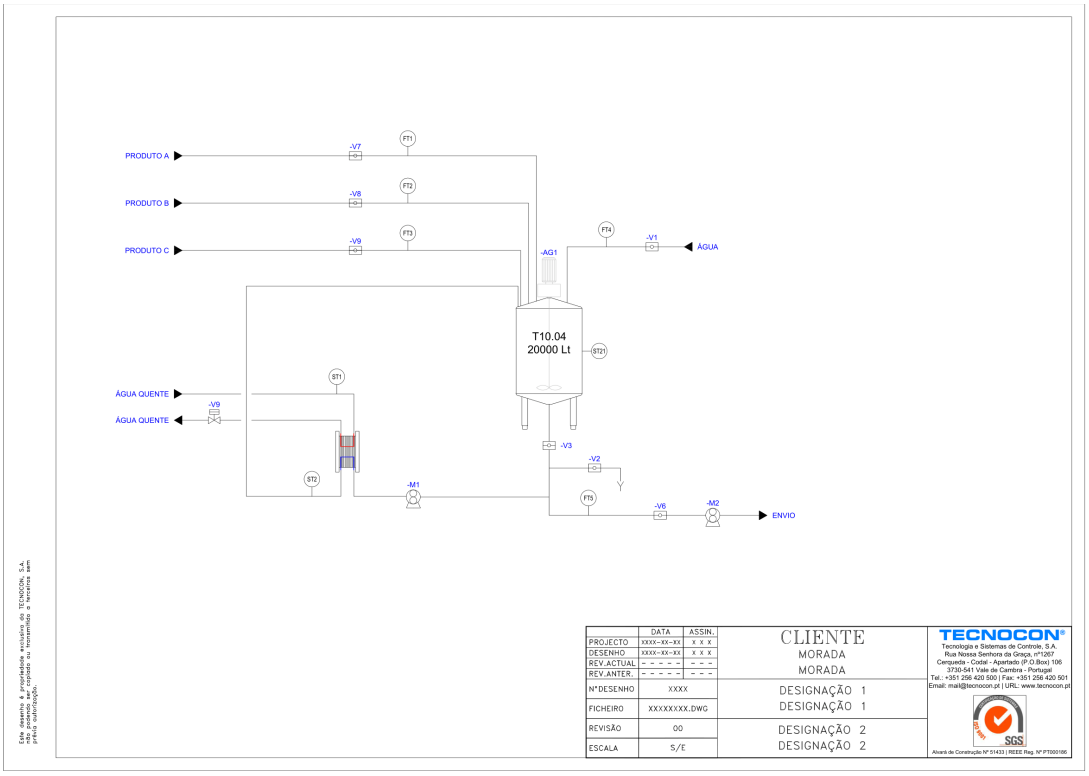


Figura 4.1: Sistema de mistura

Na Figura 4.1 está representado um esquema do projeto. Este trata-se de um sistema de mistura, em que existem três tipos de produtos e água que são direcionados para o tanque para fazer a sua mistura, com a opção de se proceder ao seu aquecimento. Após a mistura estar pronta é redirecionada para uma determinada linha. Este projeto foi desenvolvido em duas fases, em que numa primeira parte se efetuou o desenvolvimento da aplicação do PLC com HMI em TIA PORTAL e numa segunda parte a

implementação em ambiente SCADA.

4.1.1 Tia Portal

Para se proceder à programação do PLC e do HMI, foi utilizado o software da Siemens, TIA PORTAL V16 [17] , em que numa primeira fase se definiu o tipo de hardware do PLC, cartas e o HMI através da opção *"Hardware Catalog"* como se pode ver através da Figura 4.2. Como já referido acima, o PLC utilizado no hardware foi um PLC S71200.

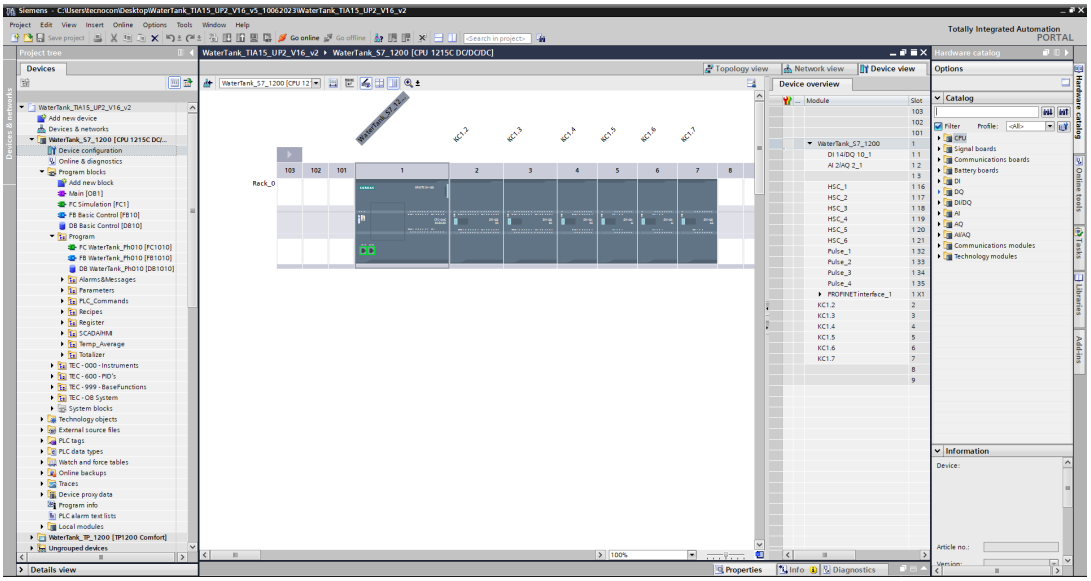


Figura 4.2: Configuração do PLC

Para haver uma comunicação entre o HMI e o PLC, estabelece-se uma ligação virtual (Figura 4.3), através do protocolo de comunicação *PROFINET*.

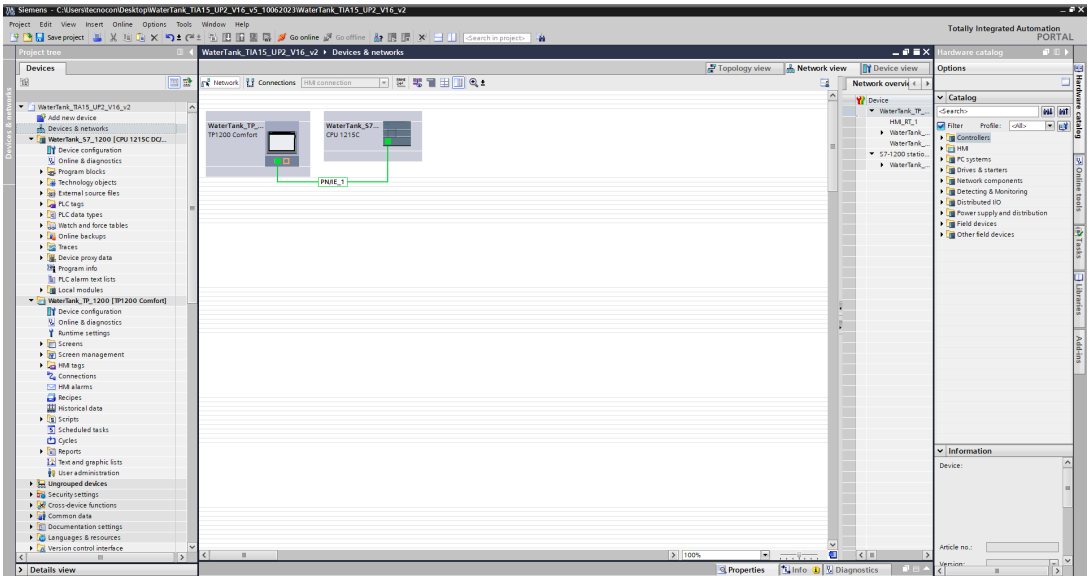


Figura 4.3: Ligação virtual entre HMI e o PLC

Para além de se estabelecer a ligação é necessário também definir os IPs para cada um dos componentes, que têm que estar na mesma gama, como se pode ver através das Figuras 4.4 e 4.5.

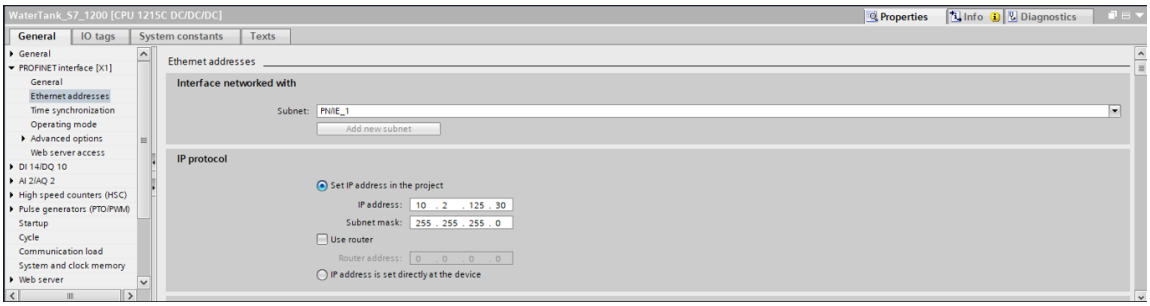


Figura 4.4: Endereço IP PLC

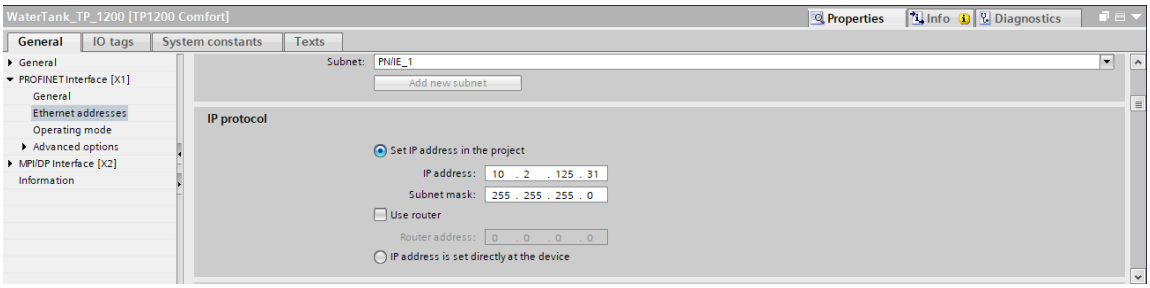


Figura 4.5: Endereço IP HMI

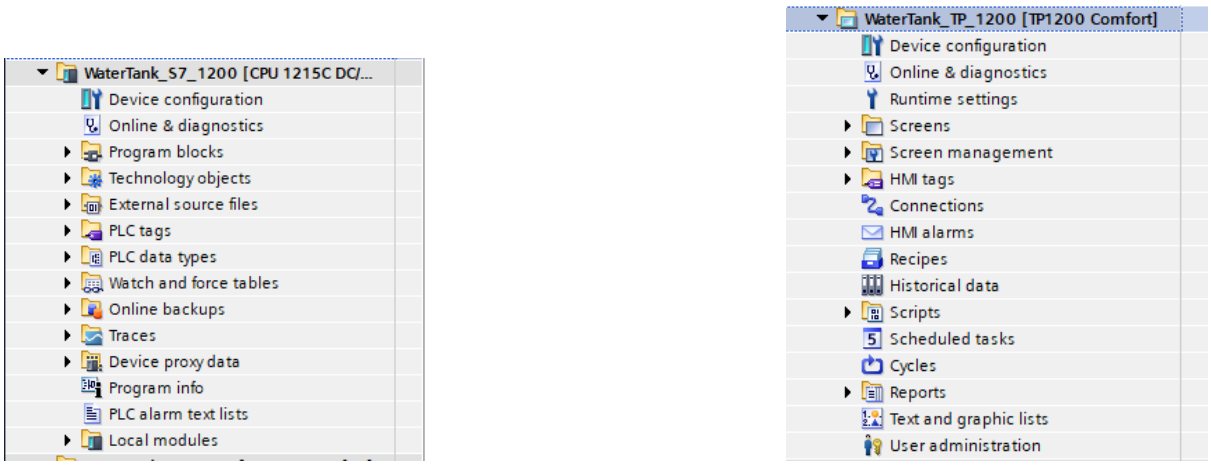
Programação

Para se proceder à programação do PLC e do HMI, estes são separados em 2 divisões no TIA PORTAL. Na área do PLC, Figura 4.6(a), são adicionadas as tags e os blocos de programação e na Figura 4.6(b) é feita a gestão das páginas da consola, alarmes, receitas e tags do HMI.

Para adicionar novas variáveis, no separador *"PLC tags"* e abaixo da última tag, insere-se uma nova tag definindo o seu *data Type* e o seu endereço (Figura 4.7). Para definir as tags na parte do HMI, basta criar a tag, definir o seu tipo, estabelecer a conexão e indicar o endereço correspondente da variável presente no PLC (Figura 4.8).

Para cada equipamento é criado um data block com todas as suas características ou propriedades associadas. Estes desempenham um papel crucial na configuração e operação adequada do sistema. Através da Figura 4.9, estão alguns atributos para uma válvula. O data block referente à válvula contempla:

- Nome da Válvula: Identifica a válvula de maneira única no sistema.
- Tipo de Válvula: Especifica o tipo de válvula, como válvula de globo, válvula de esfera, válvula borboleta, etc.



((a)) ((b))

Figura 4.6: Desenvolvimento no PLC Siemens. (a) Área PLC; (b) Área HMI

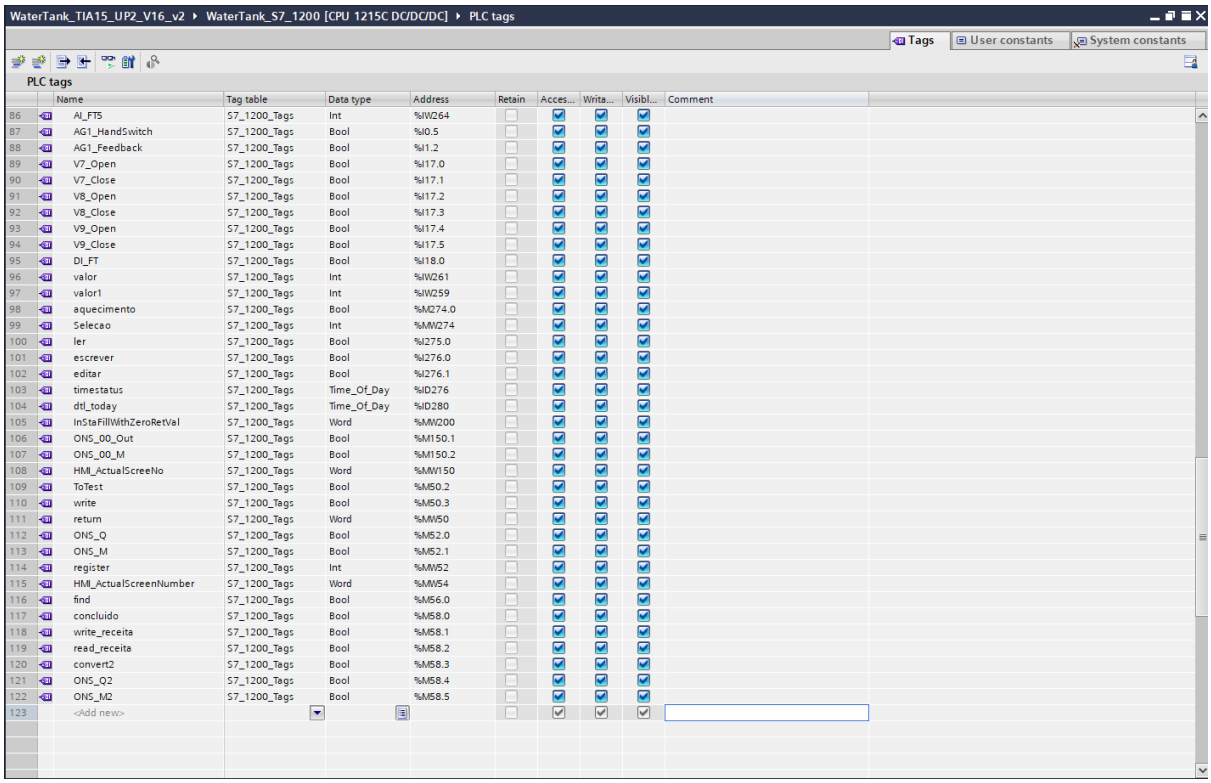


Figura 4.7: Tags PLC

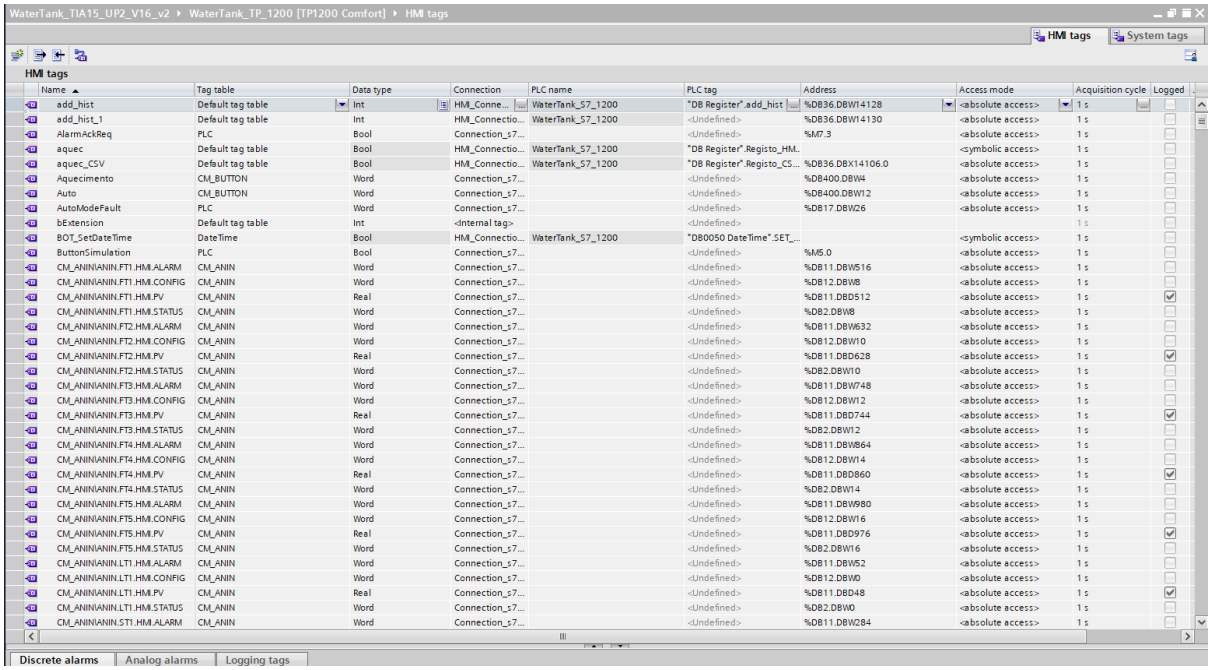


Figura 4.8: Tags HMI

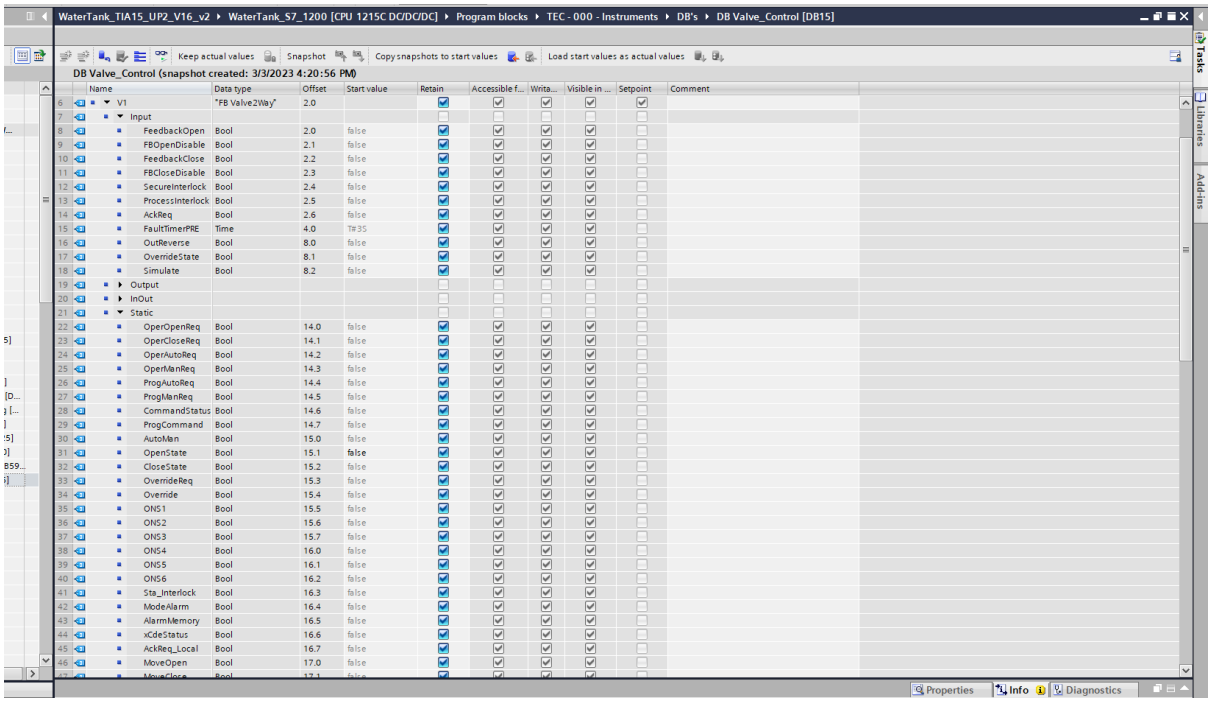


Figura 4.9: Atributos de uma válvula

- Posição da Válvula: Indica se a válvula está aberta, fechada ou em uma posição intermediária.
- Estado da Válvula: Pode fornecer informações sobre o estado atual da válvula, como se está em operação ou em manutenção.
- Alarmes e Eventos: Configurações relacionadas a alarmes e eventos associados à válvula, incluindo limites, prioridades e configurações de notificação.
- Unidades de Engenharia: Define as unidades de engenharia associadas às medições da válvula, como pressão, fluxo ou temperatura.
- Tempo de Resposta da Válvula: Indica o tempo necessário para que a válvula passe de uma posição totalmente aberta para uma posição totalmente fechada (ou vice-versa).
- Feedback da Válvula: Informações de *feedback* que indicam a posição real da válvula em comparação com a posição desejada.
- Intertravamento: Configurações relacionadas a intertravamentos de segurança para garantir que a válvula só possa ser operada sob condições específicas.

Através da Figura 4.10, estão alguns atributos para um motor (data block). As principais propriedades são:

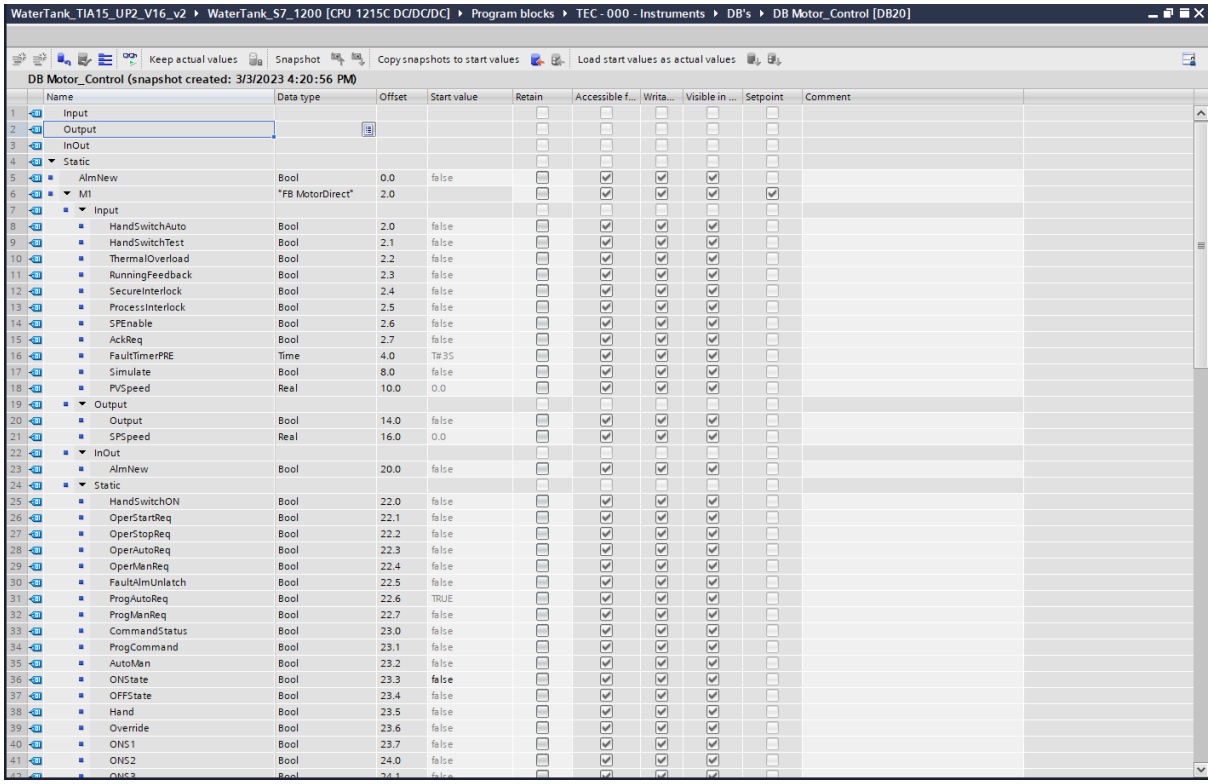


Figura 4.10: Atributos de um motor

- Nome da motor: Identifica o motor de maneira única no sistema;
- Tipo de motor: Especifica o tipo de motor, como motor direto ou motor com variador;
- Parâmetros de Controle: Configurações de controle, como velocidade desejada, limites de corrente, direção do movimento, etc
- Status do Motor: Informações sobre o estado atual do motor, como se está ligado/desligado, em movimento, em repouso, etc.
- Feedback do Motor: Dados provenientes de sensores ou encoders associados ao motor, fornecendo informações sobre posição, velocidade, temperatura, etc.
- Alarmes e Mensagens: Registo de eventos, alarmes ou mensagens relacionados ao motor.

Através da Figura 4.11, estão alguns atributos para um controlador PID, utilizado neste sistema para regular a abertura da válvula de aquecimento em função da temperatura no tanque.

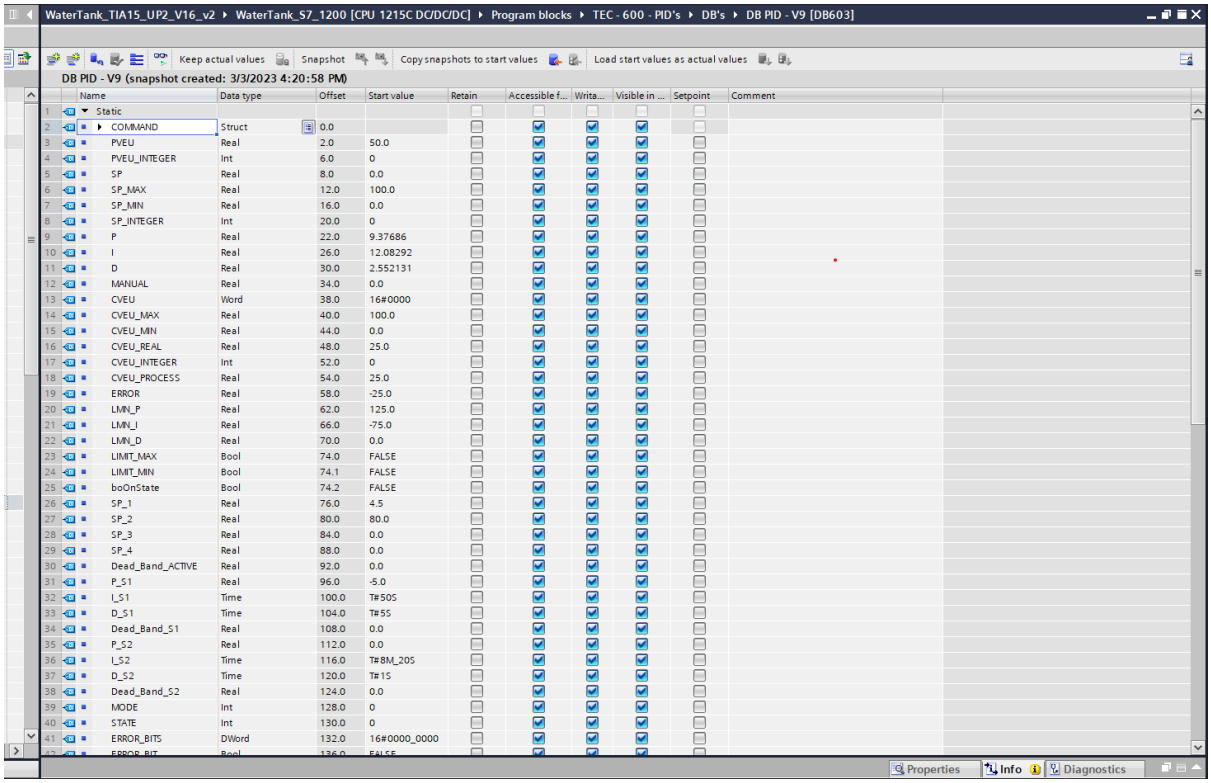


Figura 4.11: Atributos para um controlador PID

PID é a sigla para "Proporcional, Integral, Derivativo", que se refere a um tipo de controlador utilizado em sistemas de controle automático. É amplamente utilizado em sistemas de controle de processos onde é necessário manter variáveis controladas próximas a um valor desejado.

Aqui estão os componentes principais do controlador PID:

- Proporcional (P): O termo proporcional é responsável por fornecer uma resposta ao erro proporcional à magnitude do erro atual. Noutras palavras, ele ajusta a saída do controlador com base na diferença entre a variável controlada e o valor desejado.
- Integral (I): O termo integral age para corrigir o erro acumulado ao longo do tempo. Ele adiciona uma componente proporcional ao produto do erro e do tempo decorrido desde o último ajuste. Isso ajuda a eliminar erros acumulados ao longo do tempo.
- Derivativo (D): O termo derivativo reage à taxa de mudança do erro. Ele prevê a tendência futura do erro e ajusta a saída do controlador para evitar que o sistema ultrapasse o valor desejado. Isso é particularmente útil para evitar oscilações e melhorar a estabilidade.

O controlador PID é amplamente utilizado porque pode ser adaptado para uma variedade de sistemas e oferece um equilíbrio entre resposta rápida, eliminação de erros acumulados e estabilidade.

Através da Figura 4.12, estão alguns atributos para uma carta analógica. As propriedades definidas do data block são:

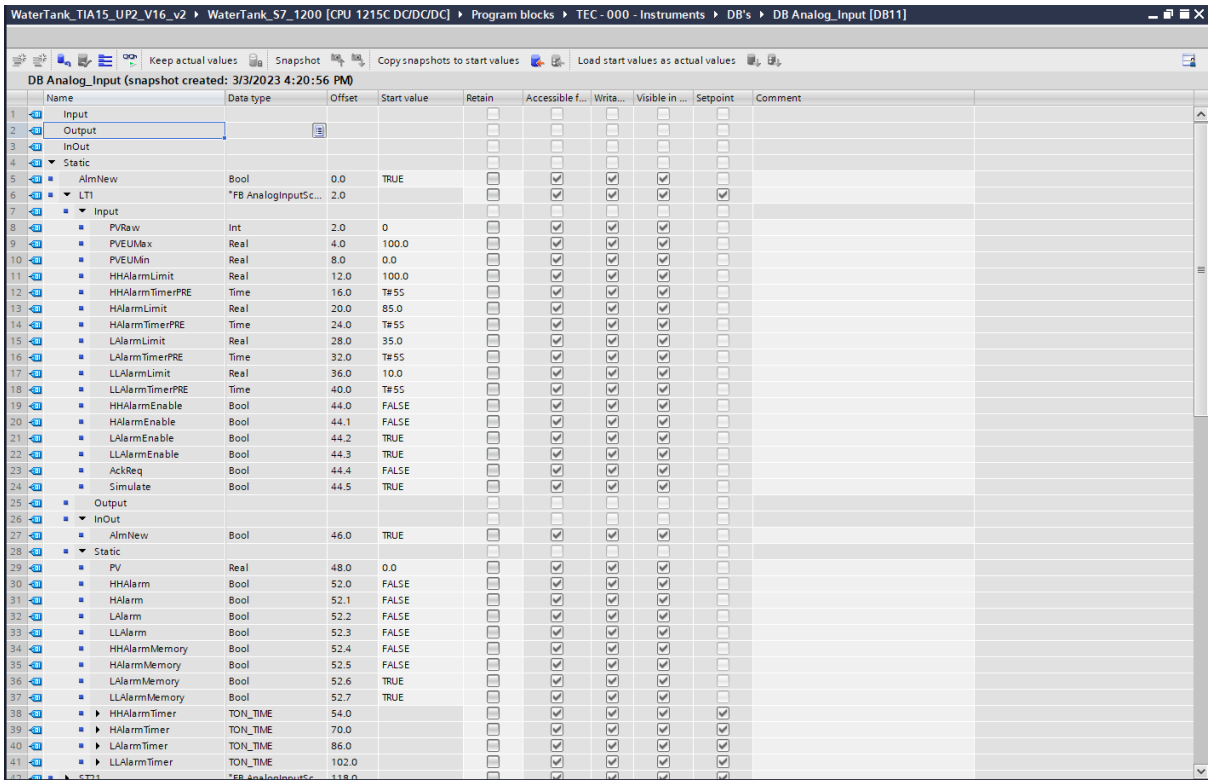


Figura 4.12: Atributos de uma carta analógica

- Variáveis Analógicas: As variáveis analógicas propriamente ditas, representando valores analógicos como temperatura, pressão, nível, etc.
- Configurações de Escala: Parâmetros que definem as escalas máxima e mínima das variáveis analógicas.
- Configurações de Engenharia: Parâmetros para configurar a engenharia de unidades (engenharia scale), como Celsius para temperatura, PSI para pressão, etc.
- Valor de Leitura Atual: O valor atual da variável analógica lido pelo PLC.
- Parâmetros de Alarme: Configurações de alarmes associados à variável analógica, como limites superior e inferior, histerese, etc.

Antes de se dar início à programação, estruturou-se um solução de programa para o respetivo sistema para uma melhor compreensão e organização.

Este processo no seu ciclo automático é composto por três fases:

- Enchimento: Esta fase, tem como objetivo colocar a quantidade de cada produto que foi parametrizada na receita, no tanque, tendo em conta a sua capacidade máxima.
- Agitação: Esta fase tem como finalidade proceder à mistura dos compostos, durante um determinado valor de tempo.
- Aquecimento: Esta etapa permite um aquecimento do produto até um valor de temperatura parametrizável e mantendo o seu valor. Esta fase é facultativa.

Para uma melhor compreensão deste sistema foi criado um fluxograma, presente na Figura 4.13, dando um exemplo de um solução para o sistema em questão, em que o seu funcionamento é predominantemente em automático, à exceção da necessidade do operador para a seleção de um novo tipo de receita ou a criação desta.

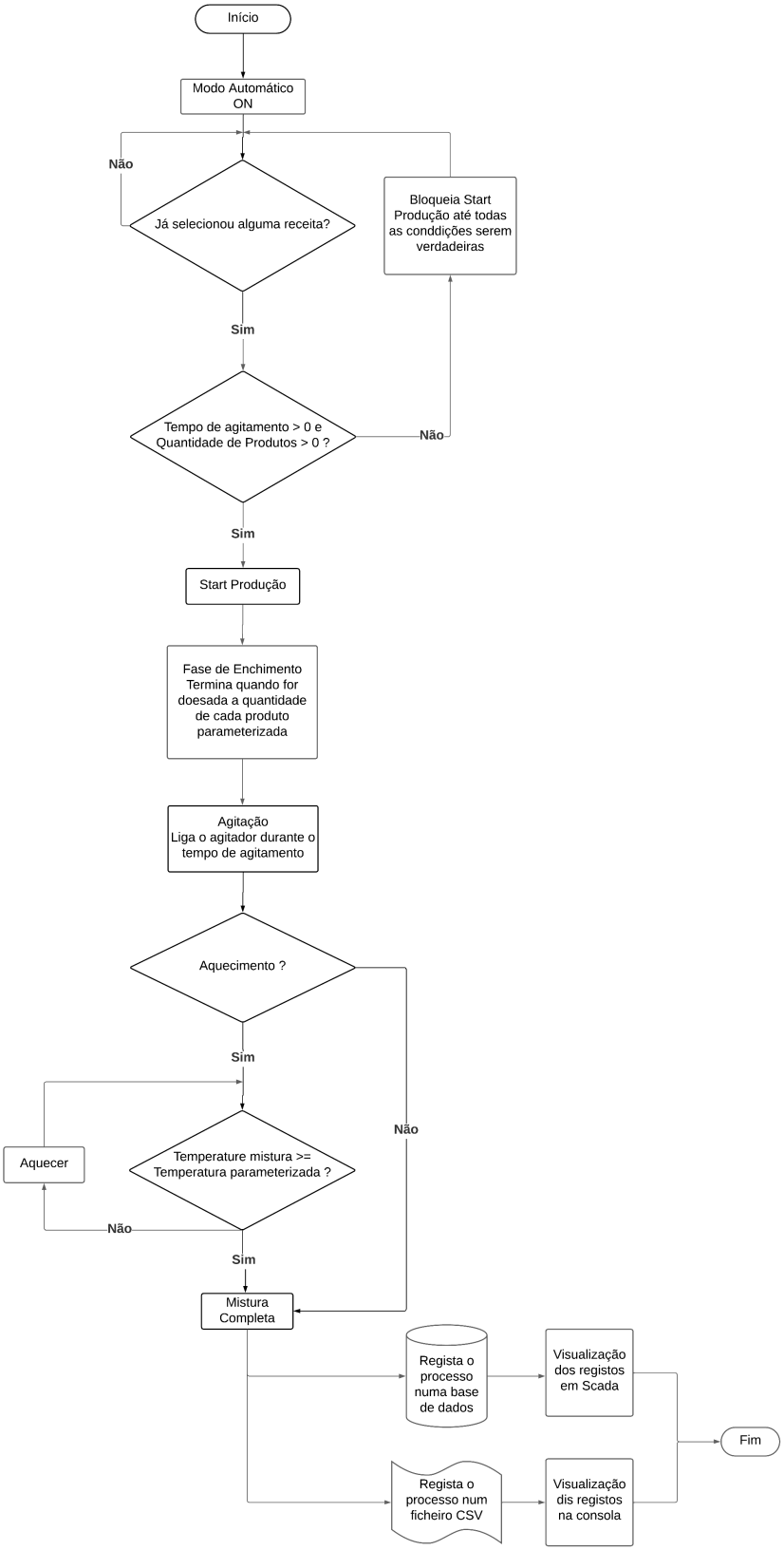


Figura 4.13: Fluxograma

4.1.2 Interface Homem-Máquina/SCADA

Estas aplicações são compostas por uma página principal, que corresponde ao sinótico da instalação, Figura 4.15 e 4.16 com várias páginas secundárias, nomeadamente, histórico de alarmes, registos, controlo, parâmetros, monitorização e autenticação. Para uma melhor compreenção foi criado um organograma, como o que se encontra presente na Figura 4.14.

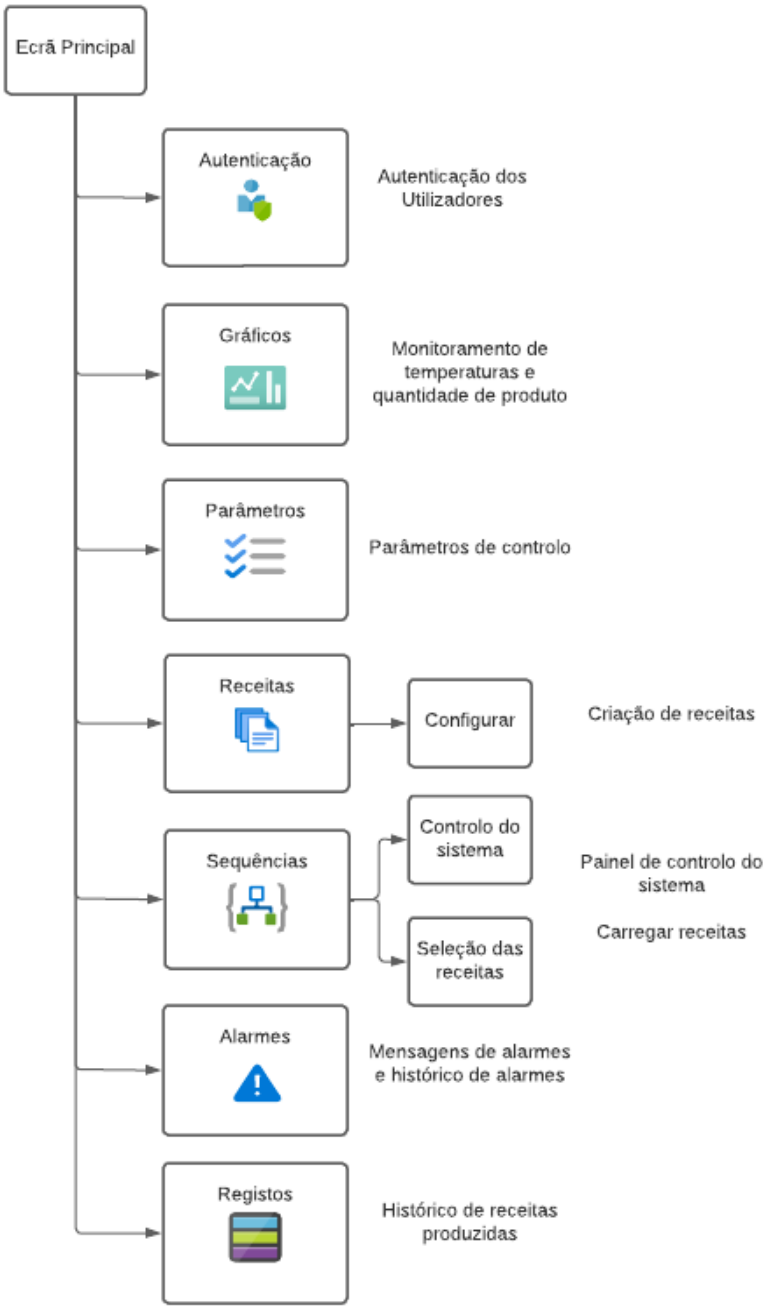


Figura 4.14: Organograma HMI

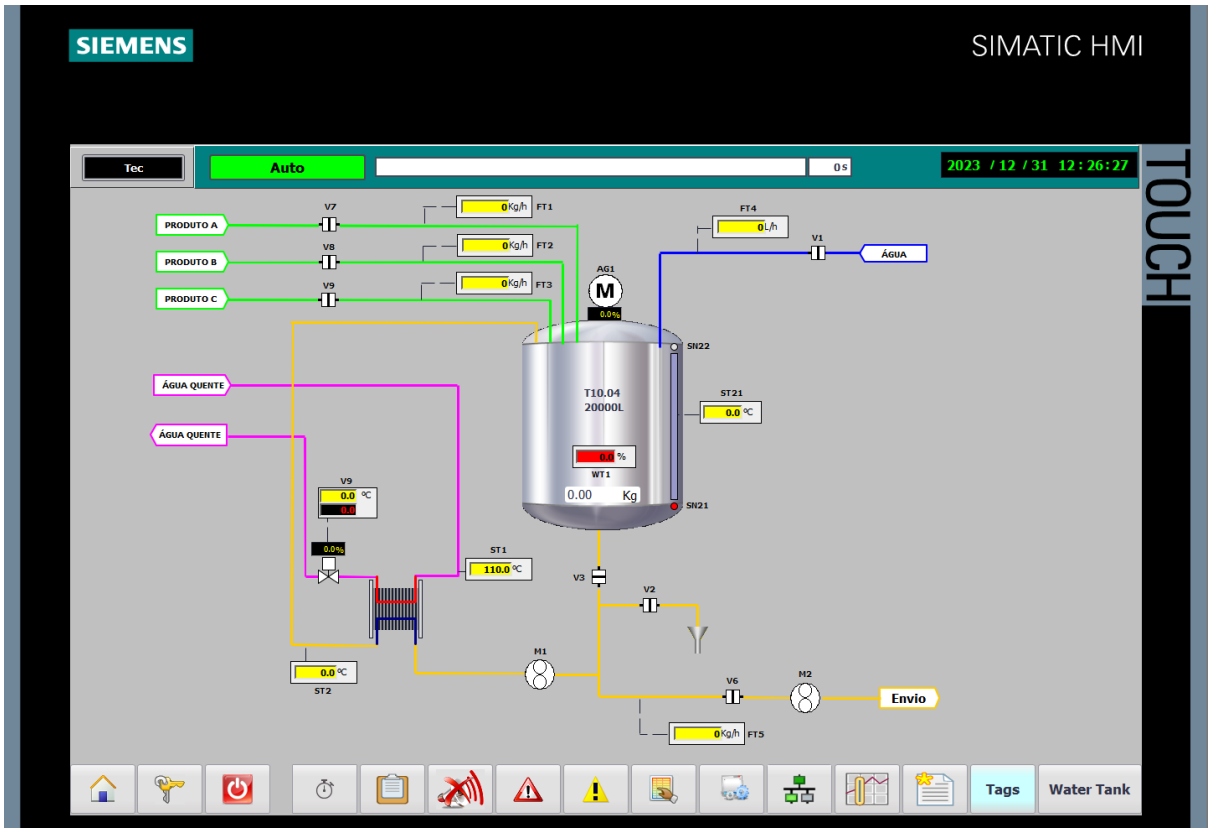


Figura 4.15: Ecrã Principal HMI, WinCC

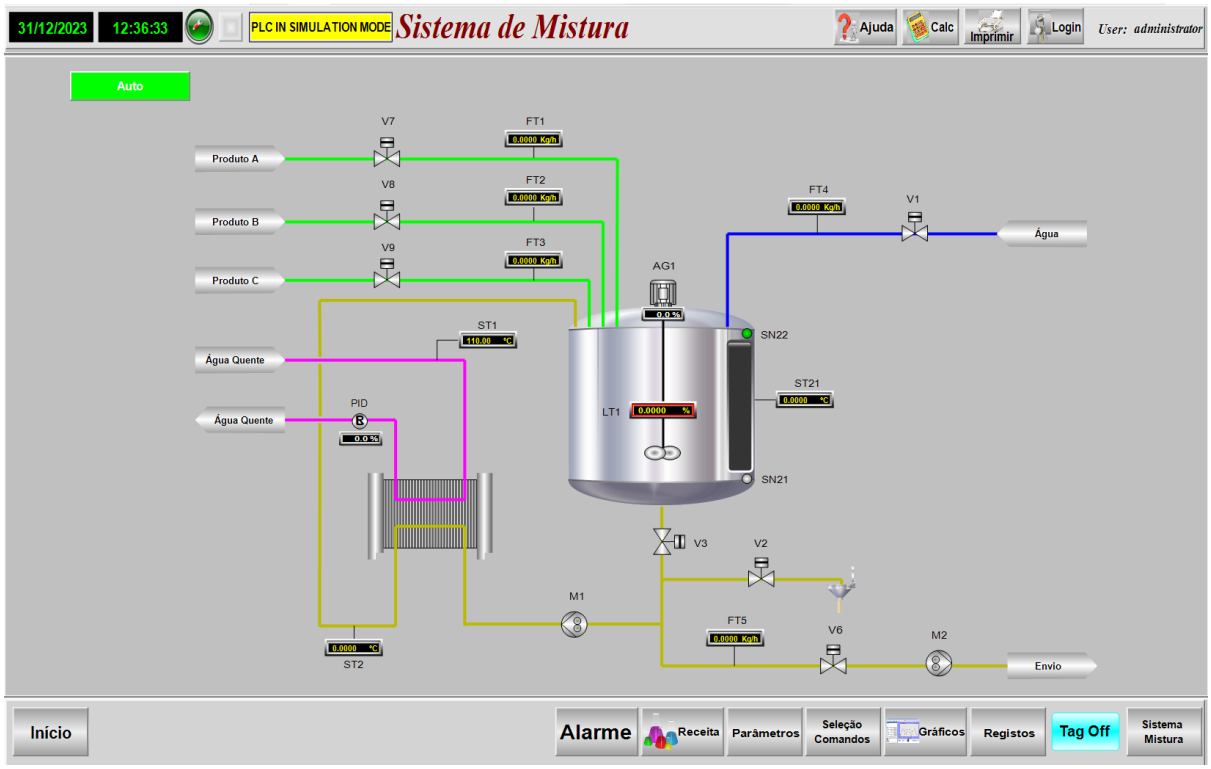


Figura 4.16: Ecrã Principal Scada, Wonderware

Receitas

O termo receitas refere-se geralmente a conjuntos predefinidos de parâmetros ou configurações associados a um determinado processo ou operação. A partir da Figura 4.17 e 4.18 temos uma visualização de uma página para criação de receitas.

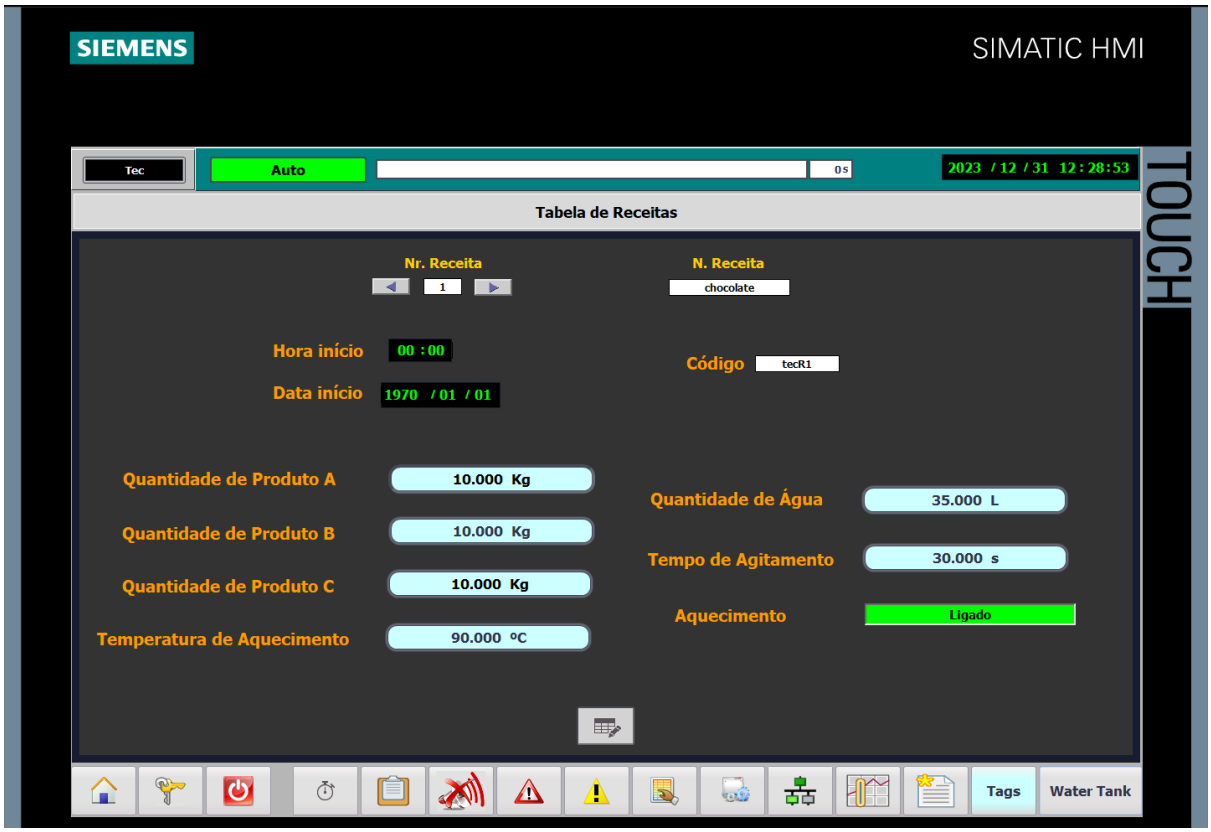


Figura 4.17: Receitas HMI, WinCC

Antes de se iniciar o processo, caso não haja a receita pretendida é necessário a sua criação, permitindo aos operadores selecionar e gerir diferentes configurações de parâmetros para um processo ou sistema automatizado. Isso pode incluir valores de set-points, limites, taxas de produção, entre outros. Esta página é usada para armazenar e recuperar conjuntos predefinidos de valores de parâmetros, que podem ser rapidamente aplicados para atender a diferentes requisitos de produção. Cada receita é estabelecida com vários parâmetros, desde a escolha da quantidade para cada produto, o tempo de agitação, temperatura de aquecimento e a escolha da opção de aquecimento. Cada receita é geralmente associada a um nome ou número identificador único para facilitar a seleção e recuperação.

Figura 4.18: Receitas Scada, Wonderware

Controlo de Processo

Para uma melhor interação do operador com o sistema, foi criada uma interface específica que exibe e controla as sequências de operações no processo industrial (Figura 4.19 e 4.20). Esta página é projetada para facilitar a monitorização, configuração e controle de sequências de trabalho ou procedimentos específicos. Indica o estado atual da sequência, como se está em execução, pausada ou concluída. Essa informação é crucial para monitorizar o progresso da operação. Botões de Controle: Inclui botões ou controles interativos para iniciar, pausar, retomar ou parar a sequência. Esses controles permitem aos operadores interagir diretamente com o processo.

Na área "Opções de Produção" são especificados 4 botões, em que cada um deles é definido por:

- Start Produção: Dá o ínicio ao processo quando o processo se encontra em automático;
- Envio: Quando o processo de produção de um determinado tipo de produto se encontra concluído, esta opção fica habilitada possibilitando ao operador fazer o arrasto do produto.
- Drenagem: Caso seja necessário esvaziar o tanque, é estabelecido o processo de drenagem habilitando a abertura das válvular V3 e V2.
- Stop Produção: É efetuada uma paragem ao processo.

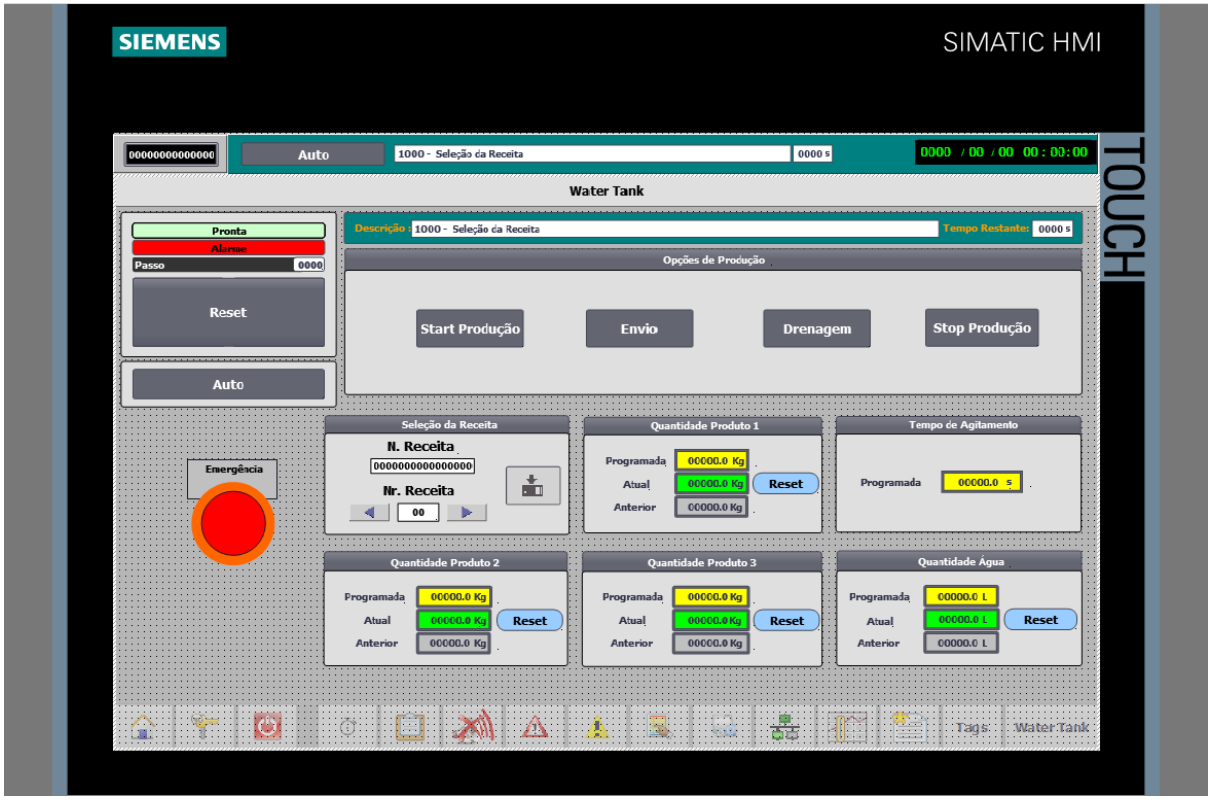


Figura 4.19: Painel de controlo HMI, WinCC

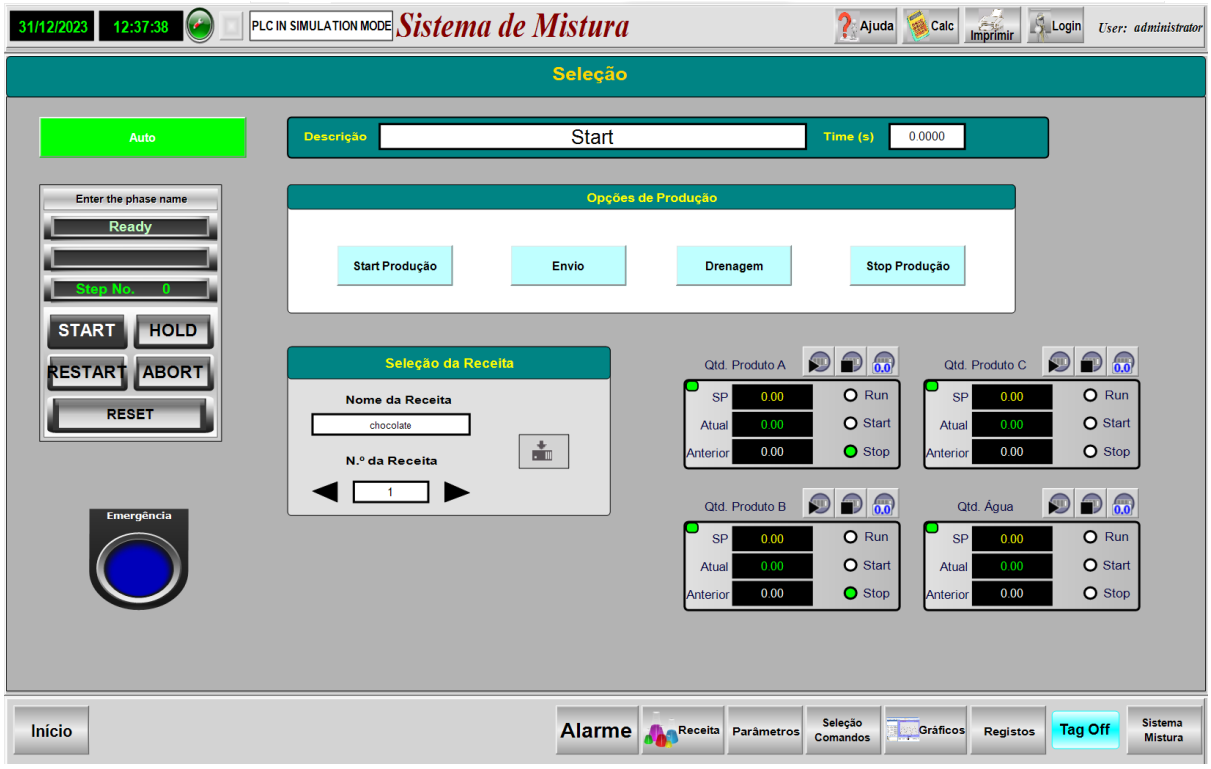


Figura 4.20: Painel de controlo SCADA, Wonderware

Na área "Seleção da receita" é onde se efetua a escolha da receita para produção através do seu número e do seu nome.

Nas áreas "Quantidade de Produto" e "Quantidade de Água" é estabelecido por 3 parâmetros:

- Programada: Nesta secção é mostrado a quantidade de produto que foi estabelecido na receita;
- Atual: Nesta secção, quando o processo se encontra em execução, é mostrado o valor da quantidade de produto que já foi doseado para o tanque.
- Anterior: Nesta secção é visualizado o valor doseado na última produção, após ser feito o reset aos valores Atual antes do começo de um novo processo.

Na área "Tempo de Agitação" encontra-se o parâmetro de tempo para a agitação do produto (Figura 4.21 e 4.22).

Parâmetros

Nesta secção são definidos alguns valores de segurança, como por exemplo setpoints de temperatura, nível do tanque e de limites de velocidade para o agitador.

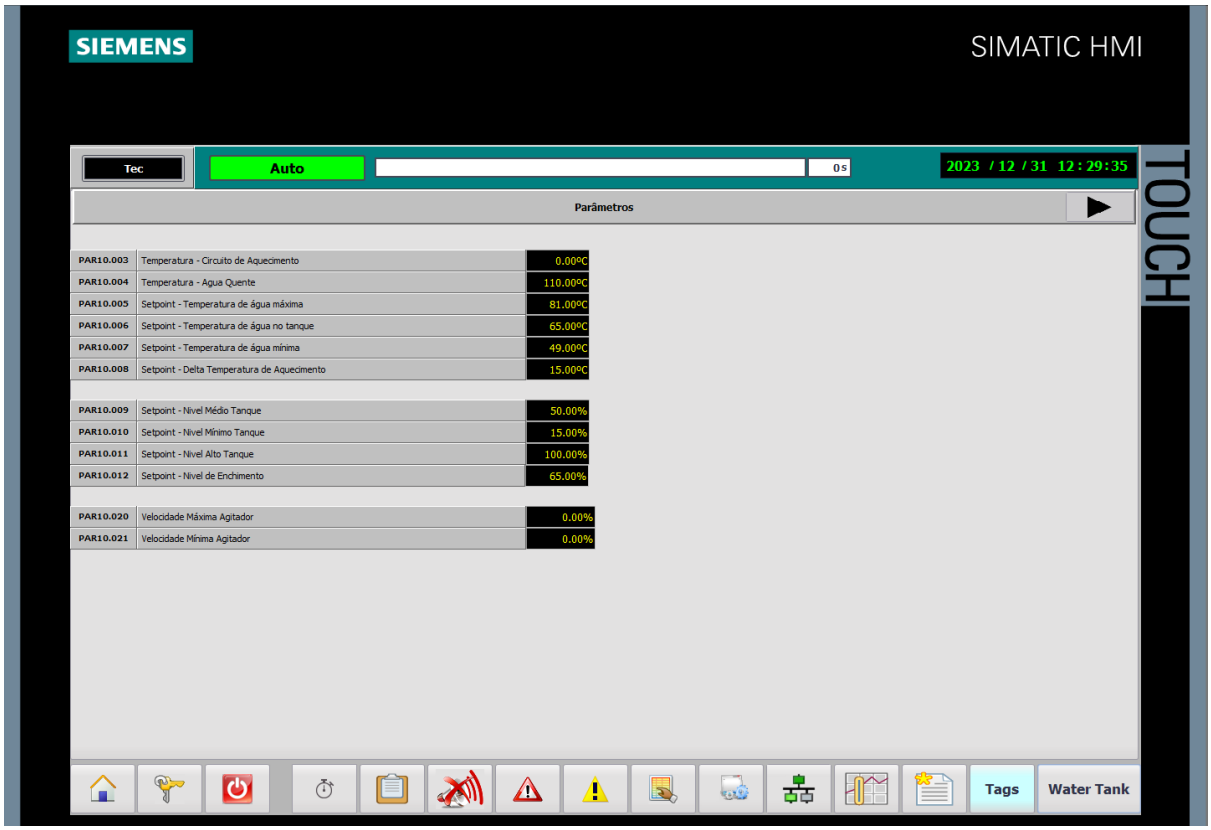


Figura 4.21: Parâmetros HMI, WinCC

Figura 4.22: Parâmetros SCADA, Wonderware

Registos

Nesta etapa (Figura 4.23 e 4.24), foi criada uma página que tem como sua finalidade mostrar um histórico das receitas que já foram produzidas, possibilitando assim uma maior supervisão dos vários tipos de produtos já produzidos. Para um acesso da consola a uma base de dados, foi criado em Visual Basic, scripts que possibilitassem a escrita destes registos no ficheiro e também a sua leitura. Estes registos são escritos num ficheiro .csv. Nos Anexos apresentam-se os códigos desenvolvidos para a escrita e leitura do ficheiro.

Como se pode observar através da tela presente na Figura 4.23 e Figura 4.24, para uma melhor procura, sabendo o intervalo de dias que se deseja obter as receitas que foram produzidas, basta que na parte superior da tela se insira a data de início e a de fim e automaticamente só vai aparecer os registos efetuados durante esse intervalo de tempo.

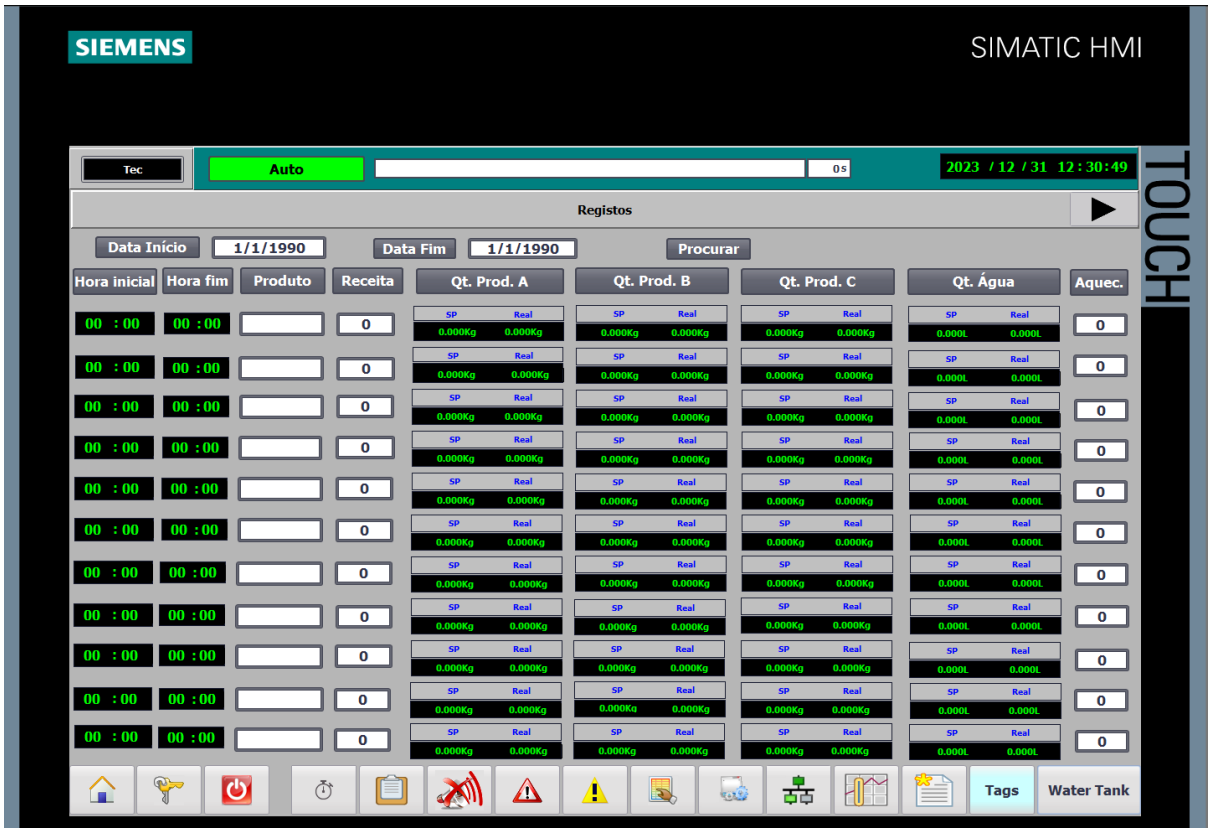


Figura 4.23: Registos HMI, WinCC

Figura 4.24: Registos SCADA, Wonderware

Histórico de Alarmes

O histórico de alarmes num sistema HMI refere-se ao registo e acompanhamento ao longo do tempo de eventos de alarmes. Estes eventos de alarmes podem incluir condições anormais, falhas, avisos ou outros eventos importantes num sistema de controle. A capacidade de rever e analisar o histórico de alarmes é essencial para a manutenção e a solução dos problemas em sistemas industriais, como se pode verificar através das Figuras 4.25 e 4.26.

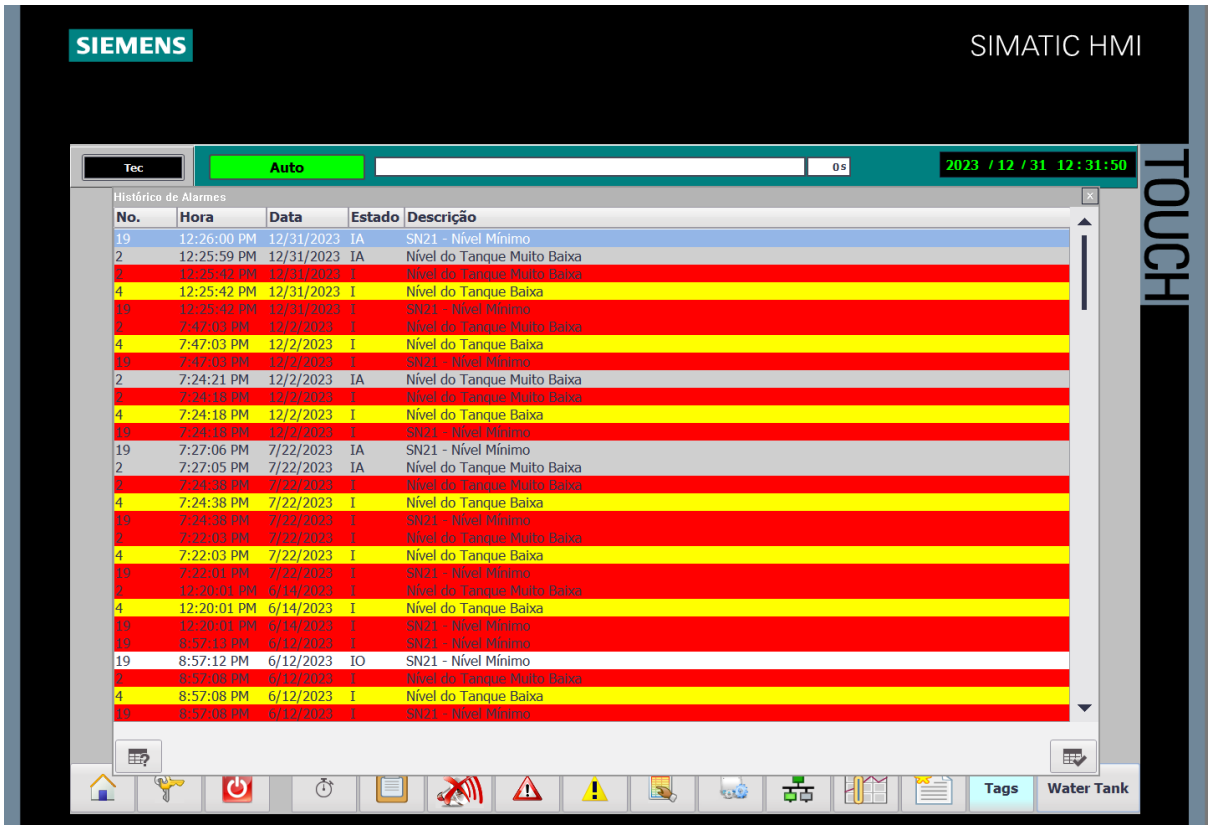


Figura 4.25: Histórico de Alarmes HMI, WinCC

Uma gestão eficaz do histórico de alarmes contribui para uma melhoria contínua do desempenho do sistema, uma redução do tempo de inatividade e uma resposta eficiente a eventos críticos. Na Figura 4.26, o histórico de alarmes é obtido através do SQL, sendo na seção 4.1.3 explicado como se desenvolveu.

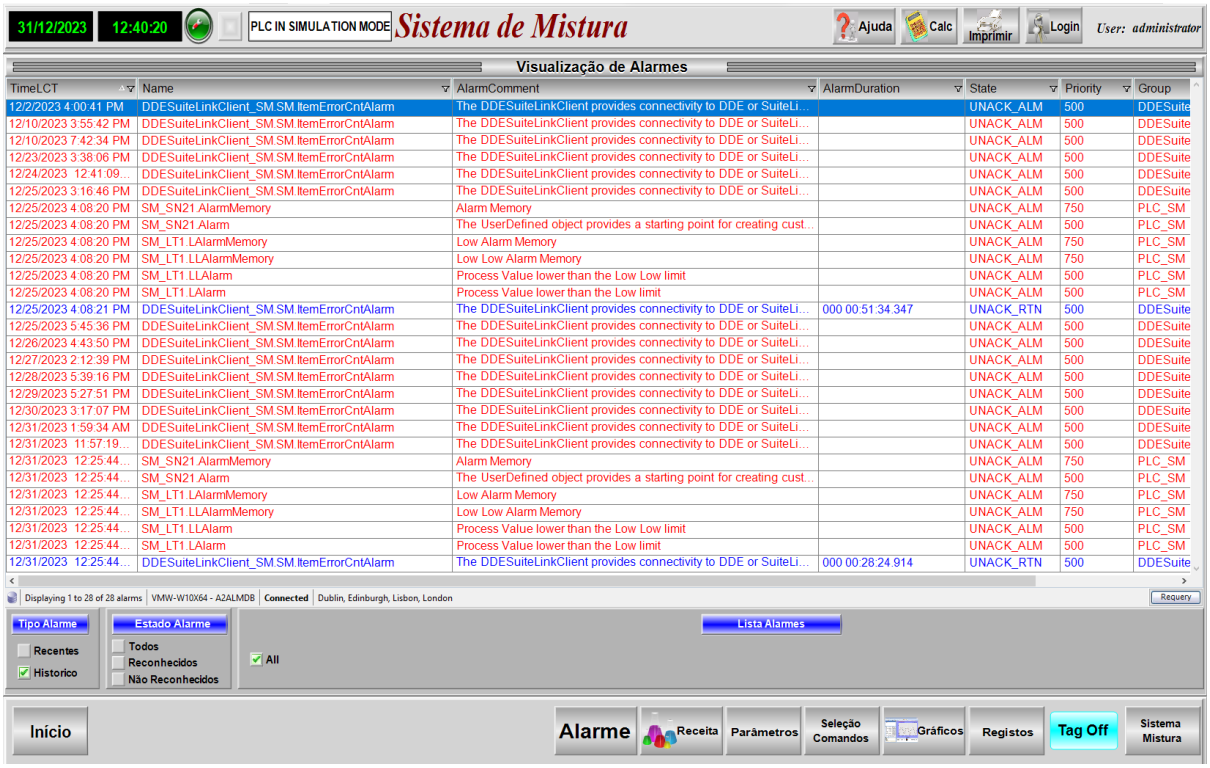


Figura 4.26: Histórico de Alarmes SCADA, Wonderware

Monitorização Visual de Variáveis

A monitorização visual de variáveis refere-se à exibição gráfica de informações sobre o estado das variáveis específicas num sistema de automação industrial. Esta funcionalidade é crucial para permitir que os operadores, engenheiros e outros utilizadores visualizem, compreendam e reajam rapidamente às condições do processo.

Para isso, como é mostrado na Figura 4.27, todo o histórico dos valores obtidos através dos sensores analógicos (nível e de temperatura) podem ser visualizados a partir desta página, proporcionando uma interface eficaz para monitorizar e controlar processos industriais, contribuindo para uma eficiência operacional e maior segurança ao sistema. No caso da Figura 4.28, esta monitorização é feita através do *Historian* sendo que só temos que habilitar esta opção no AppEngine da nossa galáxia.

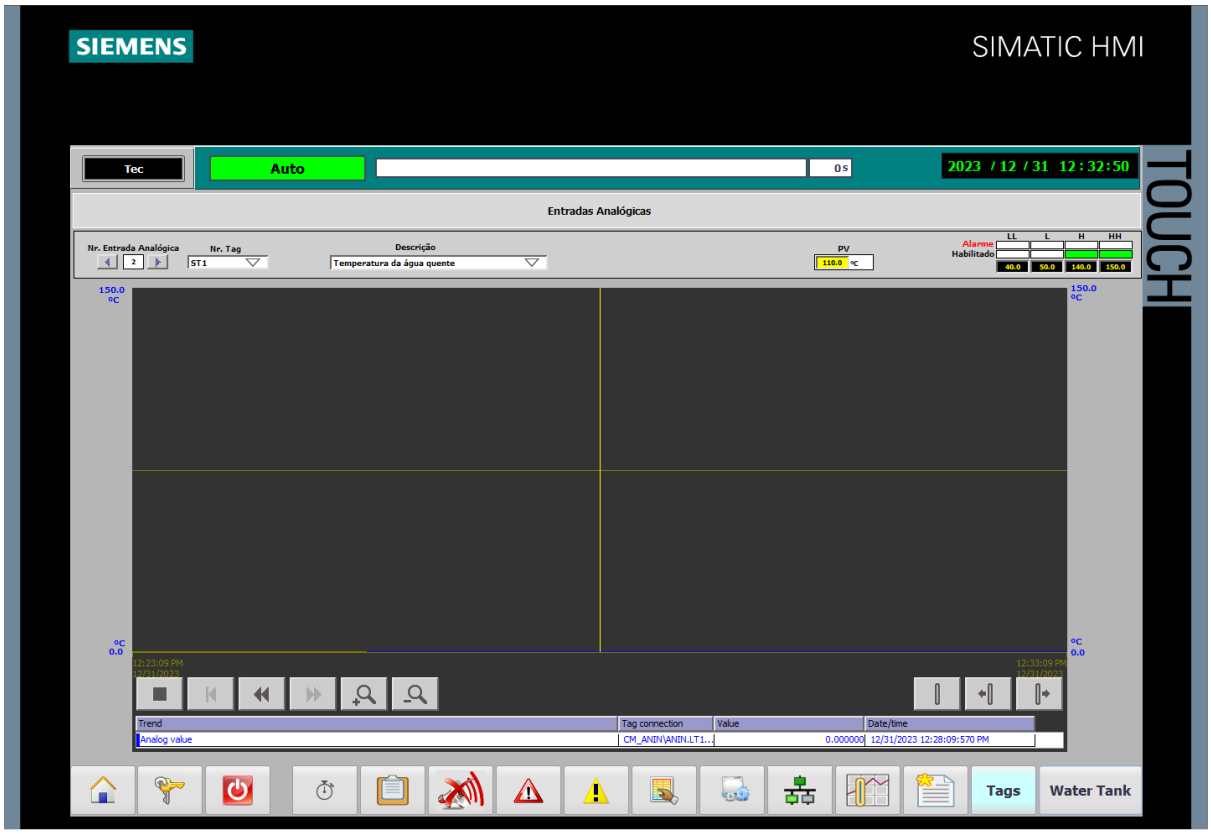


Figura 4.27: Monitorização de variáveis HMI, WinCC

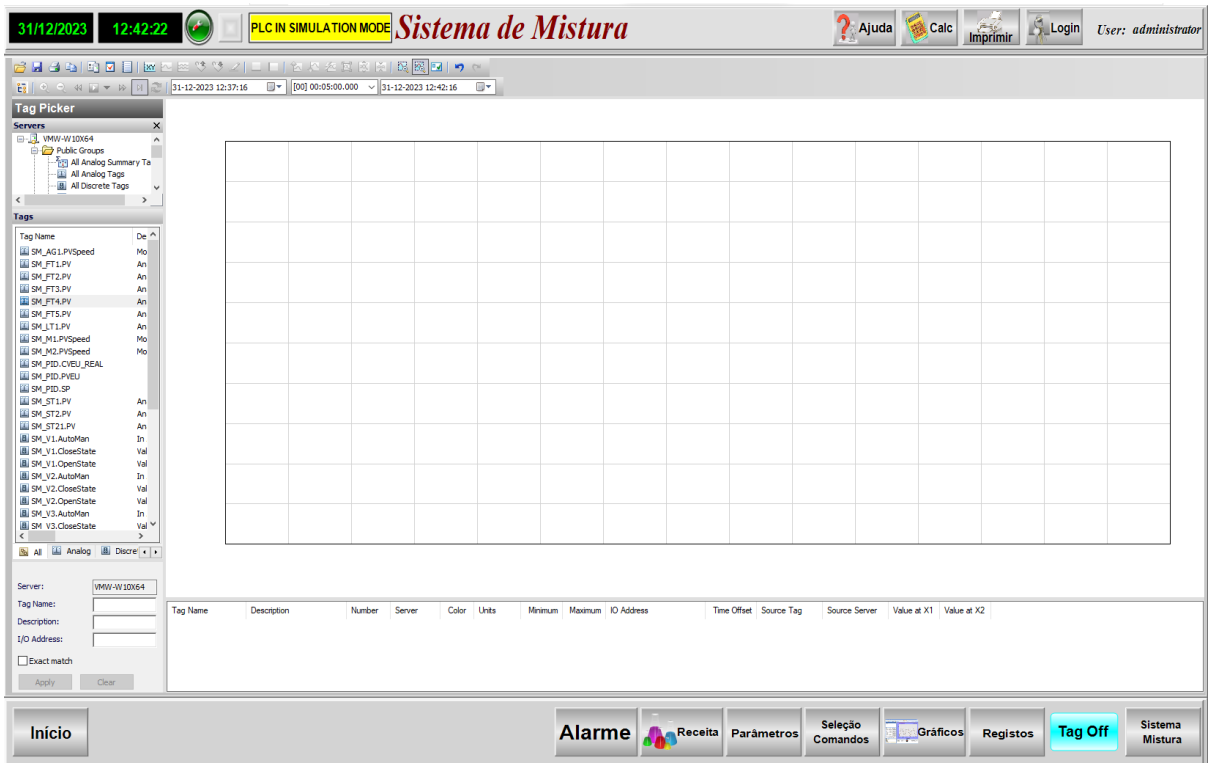


Figura 4.28: Monitorização de variáveis SCADA, Wonderware

4.1.3 Wonderware

Comunicação com Siemens

Para poder haver comunicação entre equipamentos Siemens e o Scada da Wonderware é necessário a utilização de uma ferramenta, cujo nome é NetToPLCSim em que permite simular uma conexão S7-PLC (PLC Siemens) usando um ambiente de simulação. Esta ferramenta é frequentemente usada em conjunto com o software de programação STEP 7 da Siemens. Ao usar o NetToPLCSim, é possível simular a comunicação entre um PLC e outros dispositivos num ambiente virtual, o que pode ser útil para testes e depuração de lógica de controle sem a necessidade de hardware PLC físico. Geralmente, a configuração do NetToPLCSim envolve a criação de uma interface de comunicação entre o software de simulação e o ambiente de programação do PLC (Figura 4.29).

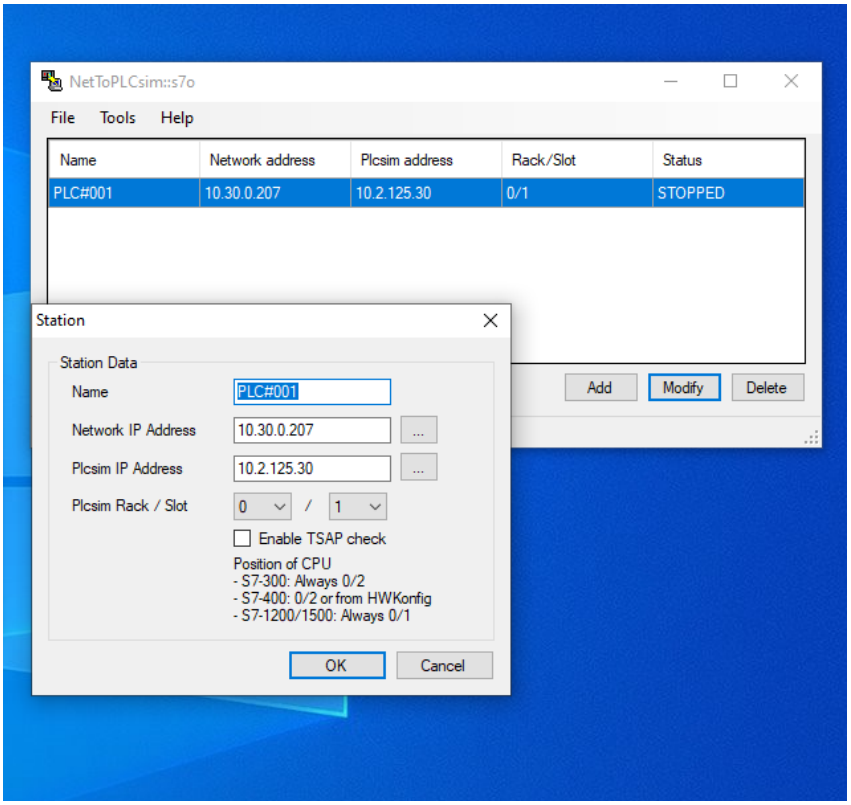


Figura 4.29: Aplicação de comunicação Nettoplcsim

Para configurarmos a ligação, basta ir à opção *"Add"* e introduzir o endereço IP da rede e o endereço IP que foi configurado no PLC previamente. No campo Rack/Slot basta seguir com a informação que se encontra imediatamente abaixo, dependendo do CPU que estejamos a utilizar. Após estas configurações, basta só dar o *Start* para podermos ter comunicação.

Para se poder começar o desenvolvimento de aplicações Scada, é necessário a criação de uma galáxia. Esta irá ser a nossa aplicação para supervisão. Para criar basta atribuirmos um nome e selecionar a opção *Create* (Figura 4.30).

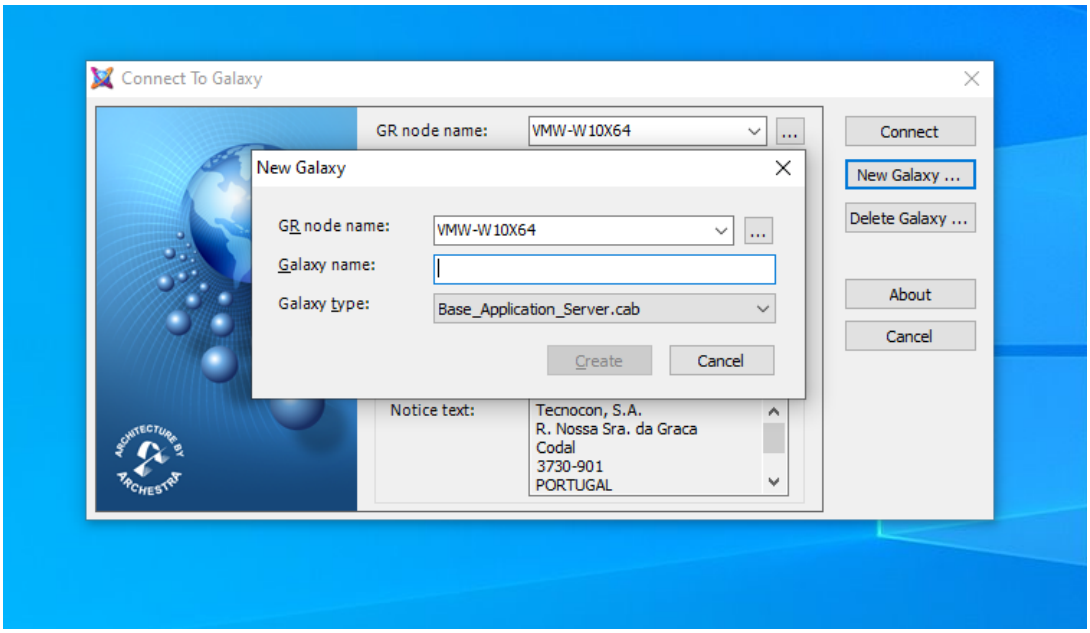


Figura 4.30: Criação da galáxia

Após a criação da galáxia, é necessário criar e configurar a interface de comunicação usada pela Wonderware para se comunicar com controladores e dispositivos industriais da Siemens, o SIDIRECT. Para isso, basta aceder ao System Management Console e na galáxia adicionar o driver de comunicação. Neste adicionou-se o endereço IP do dispositivo (Figura 4.31).

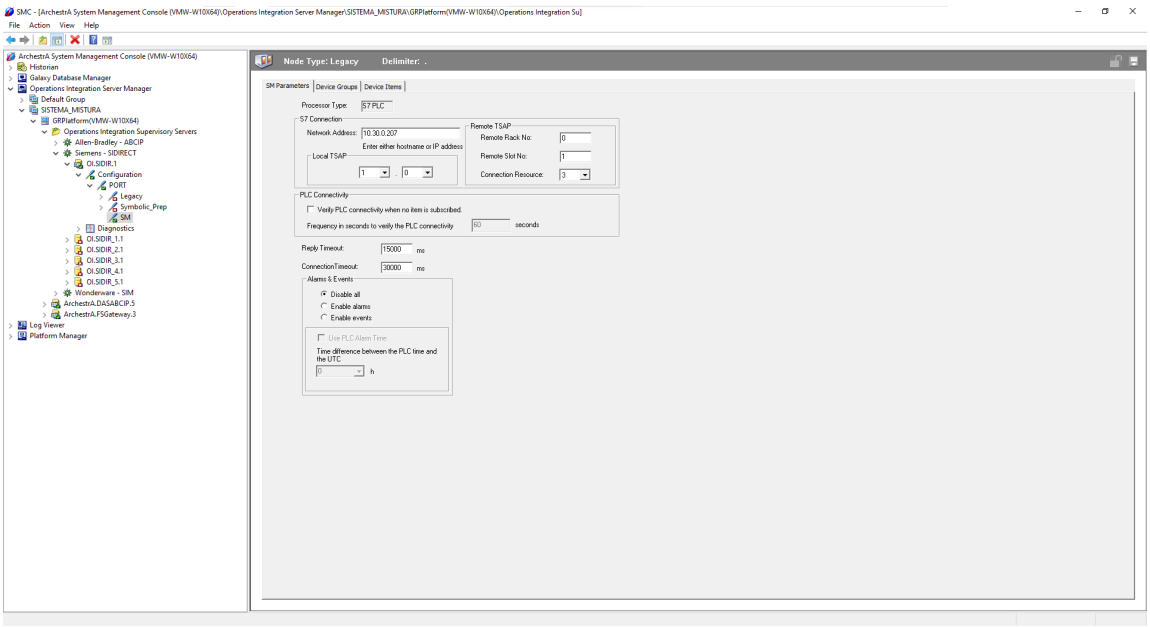


Figura 4.31: Driver de comunicação

No separador *Device Groups* inseriu-se um tópico que vai funcionar como um elemento de ligação entre as variáveis criadas no PLC para os objetos do Scada (Figura 4.32).

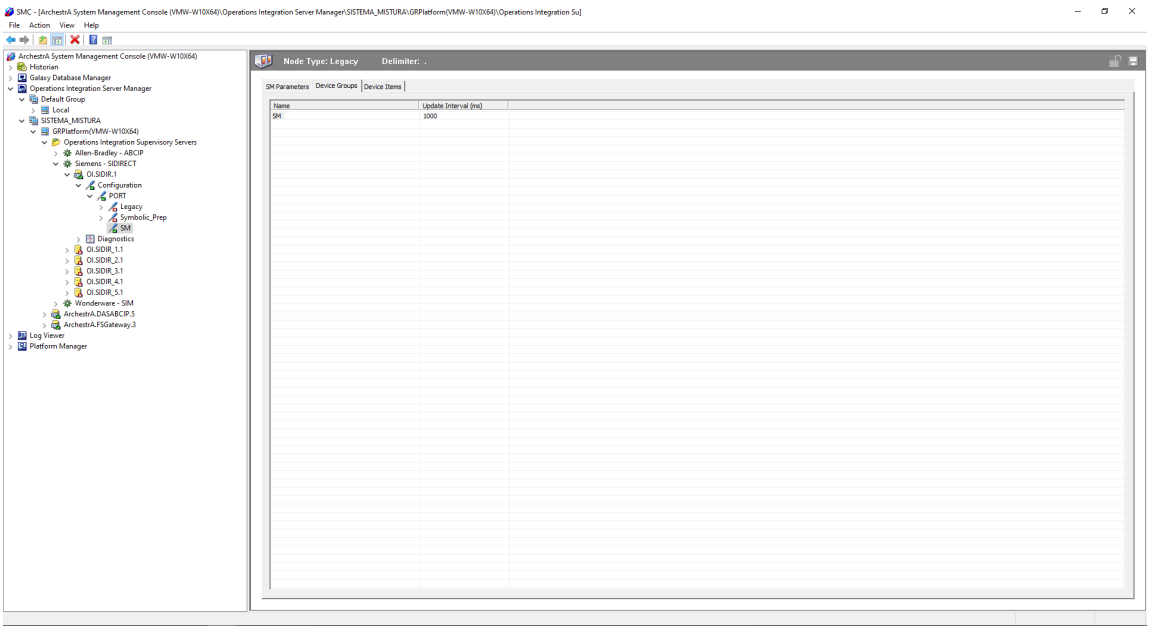


Figura 4.32: Tópico usado para a comunicação

Após a criação, este será o ambiente de desenvolvimento da nossa aplicação (Figura 4.33). É constituido por 2 toolbox, template toolbox e graphic toolbox. No contexto do ArchestrA [2], os termos "model", "deployment", e "derivation" referem-se a conceitos importantes relacionados ao design, implementação e manutenção de sistemas de automação industrial.

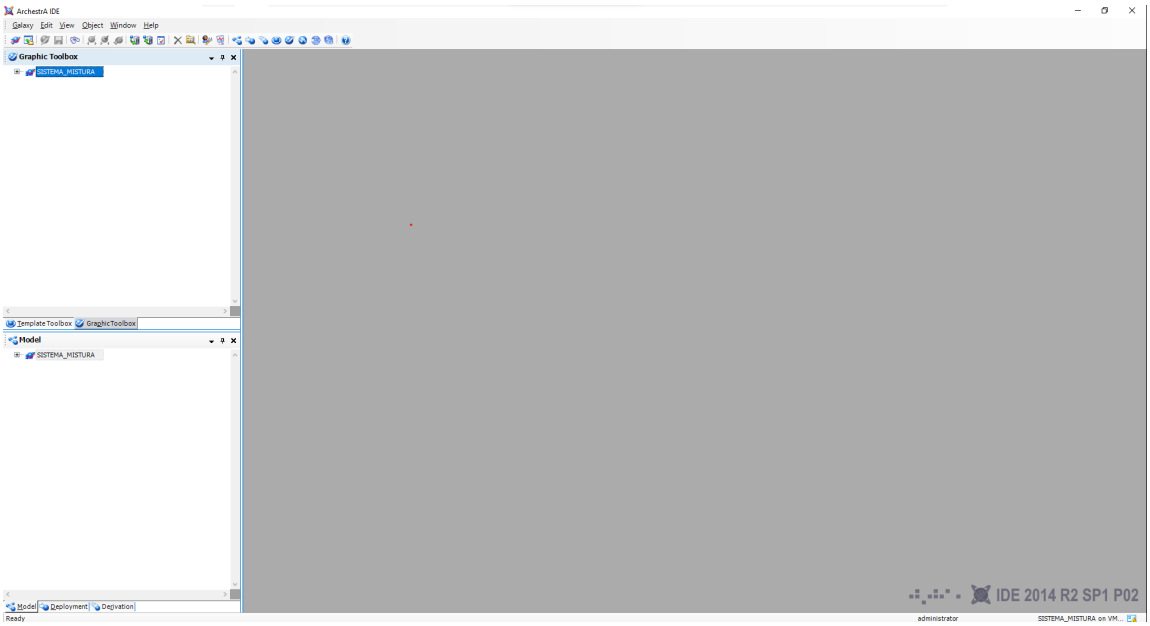


Figura 4.33: ArchestrA IDE [2]

A *Template toolbox* (Figura 4.34) corresponde a um conjunto de modelos ou objetos pré-construídos que podem ser utilizados como ponto de partida para a criação de tipos específicos de aplicações industriais. Estes objetos podem ser dos mais variados tipos, desde (por exemplo, equipamentos, bombas, válvulas, etc.).

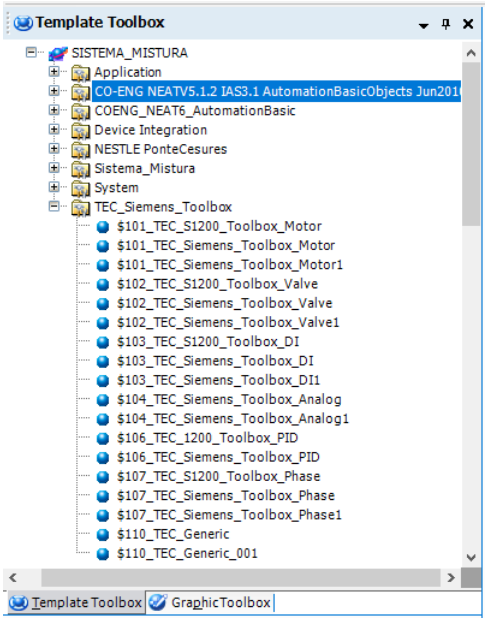


Figura 4.34: Template toolbox

A *Graphic toolbox* (Figura 4.35) corresponde a uma coleção de ferramentas gráficas disponíveis no software para criar interfaces visuais. Estas ferramentas são usadas no Wonderware InTouch, que é a interface gráfica de operador (HMI - Human-Machine Interface) do ArchestrA, para projetar e visualizar sistemas de controle industrial.

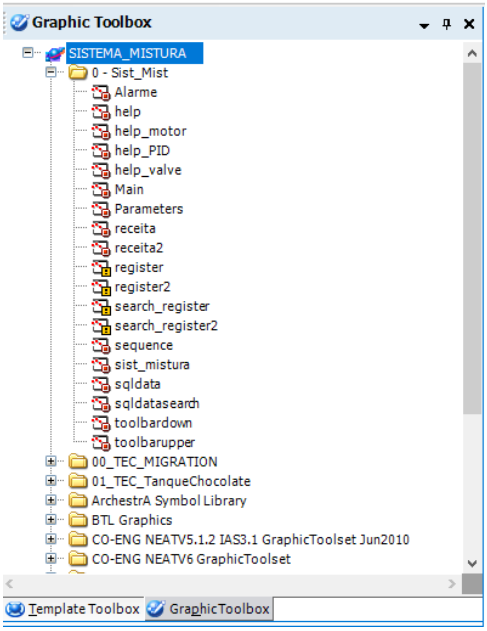


Figura 4.35: Graphic toolbox

O *Model* (Figura 4.36) como blocos de construção para criar instâncias específicas de objetos em um sistema de automação industrial. Eles são a base para padronizar o comportamento e as características de objetos semelhantes.

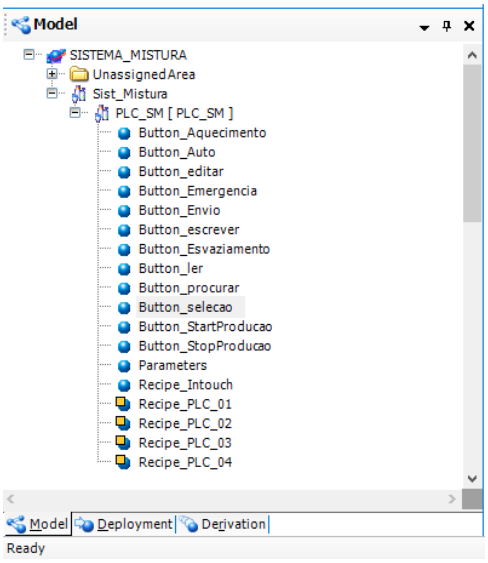


Figura 4.36: Model

No *Deployment* (Figura 4.37) os modelos são associados a equipamentos específicos ou processos industriais para criar instâncias reais desses objetos. Isso inclui a configuração de parâmetros específicos para adaptar o modelo à operação real.

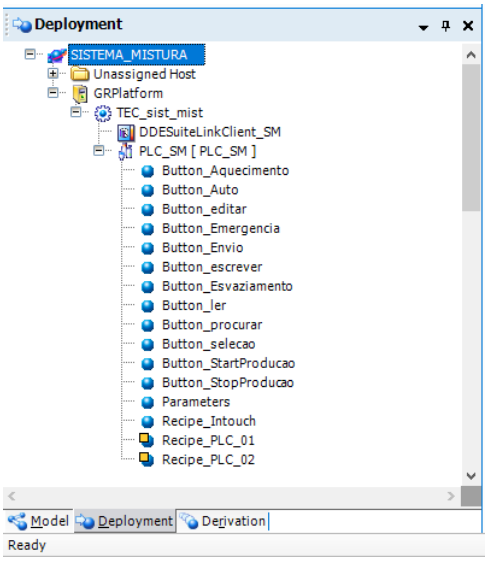


Figura 4.37: Deployment

O *Derivation* (Figura 4.38) é útil para criar variações de um modelo básico sem ter que redesenhar completamente o novo modelo. Isso facilita a manutenção e a atualização, pois as mudanças no modelo principal podem ser refletidas nos modelos derivados.

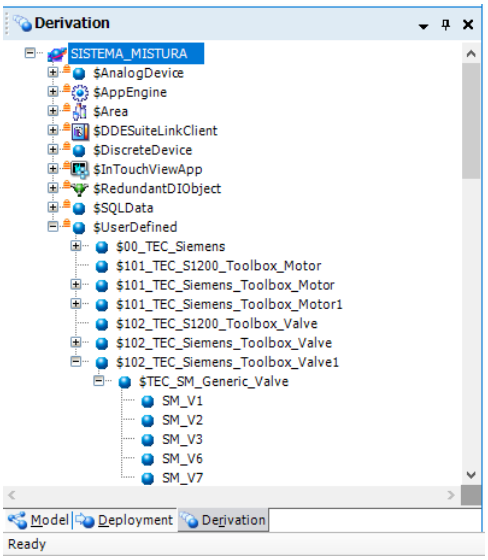


Figura 4.38: Derivation

Através da Figura 4.37 podemos ver a estrutura da galáxia, que é formada através do GRPlatform onde é feito o "deploy" da galáxia, `"TEC_siat_mist"` trata-se do "engine" e `"PLC_SM"` corresponde à área do processo, em que nela se encontram todos os objetos que foram criados para a aplicação.

Para se perceber melhor o que é o engine do archestrA (Figura 4.39), pode comparar-se como o motor do carro - ele "alimenta" tudo. A taxa de verificação determina a taxa de atualizações de dados, a velocidade de processamento de alarme, a velocidade de historização e a taxa de execução de scripts. Nada é executado mais rápido do que a taxa de verificação do appengine. Uma plataforma também pode ter vários mecanismos de aplicativos. Se houver vários objetos (Figura 4.40) que atualizem com taxas diferentes, poderá configurar-se mecanismos independentes na mesma plataforma com taxas de verificação diferentes.

O mecanismo também controla o historian, pois contém a localização configurada do histórico. Qualquer objeto configurado com histórico, usará esta função, como por exemplo um histórico de alarme.

Este também controla a redundância. Quando configurado corretamente, se a plataforma para a qual um mecanismo é implantado ficar offline, um mecanismo redundante pode assumir o processamento de todos os objetos hospedados, garantindo perda mínima de dados - se houver.

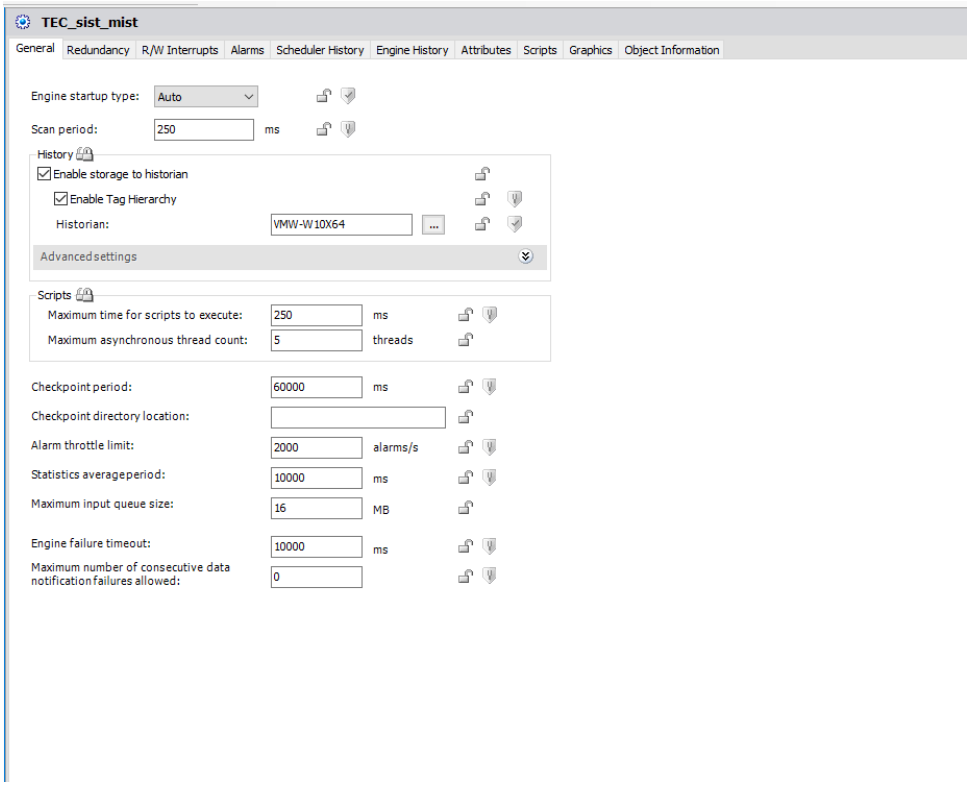


Figura 4.39: AppEngine

Para fazermos a ligação entre a área e o nosso protocolo de comunicação é necessário a configuração de um atributo, "DIOBJECTPATH", em que vai lhe é atribuído o caminho `"DDESuiteLinkClient_SM.SM"` (Figura 4.40) .

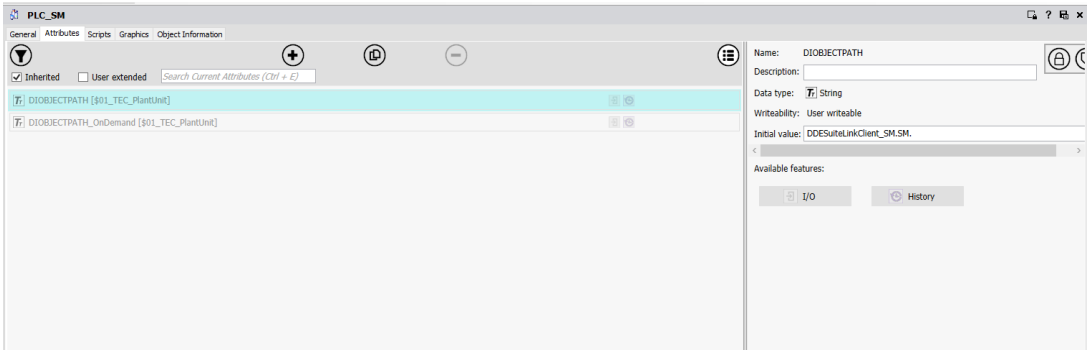


Figura 4.40: Diobjectpath

Este atributo vai ser necessário para podermos estabelecer o caminho dos vários atributos para cada objeto, como vai ser mostrado mais à frente.

O objeto DDESuiteLinkClient fornece uma conexão de cliente com servidores DDE ou SuiteLink para que os dados possam ser lidos e gravados em fontes de dados externas, como um PLC. Para o criar basta criar uma instância, como na Figura 4.41 e 4.42.

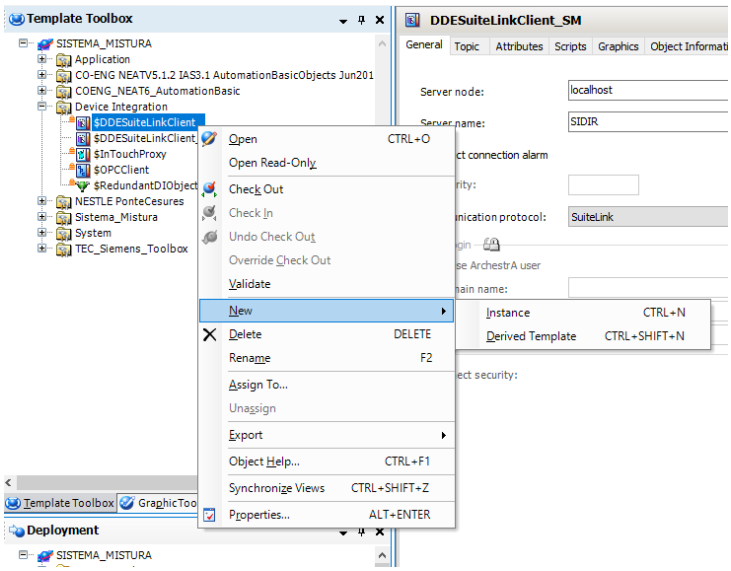


Figura 4.41: Criação do protocolo DDESuiteLink

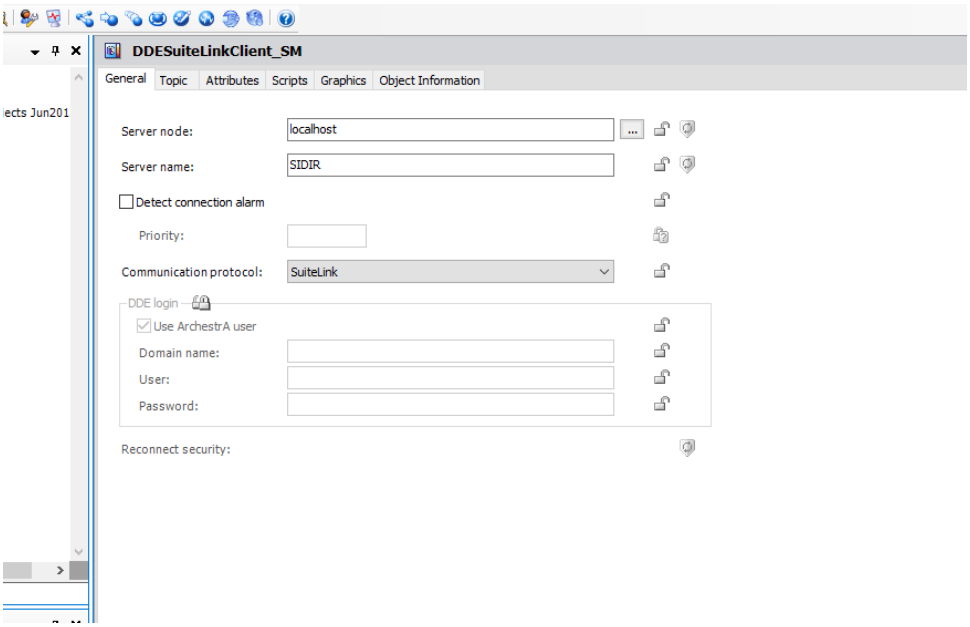


Figura 4.42: Configuração do protocolo DDESuiteLink

Assim temos:

- "Server node" – configurado com o nome do nó ou do endereço IP onde a OI ou DAServer está localizada (neste exemplo, o servidor OI está localizado no host chamado localhost);
- "Server name" – configurado com o nome do servidor DDE ou Suitelink (neste exemplo, o nome do aplicativo usado é SIDIR;

O tópico tem que corresponder exatamente ao tópico ou grupo de dispositivos que foi configurado na OI ou no DAServer no SMC (Figura 4.43).

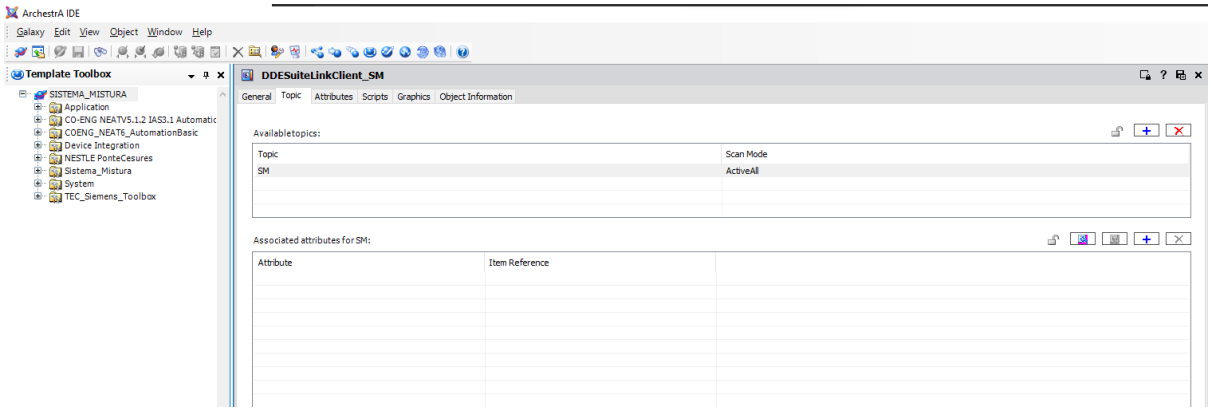


Figura 4.43: Tópico de comunicação

Para cada objeto, seja válvula, motor, etc..., estes são compostos por várias características ou propriedades associadas a esse equipamento específico. Como estes atributos se encontram presentes no PLC, como já foi mostrado através da Figura 4.9 para uma válvula, no ArchestrA estes também têm que ser criados para podermos fazer a ligação com o objeto gráfico. Na Figura 4.44 estas caraterísticas são inseridas no separador *Field Atributtes*.

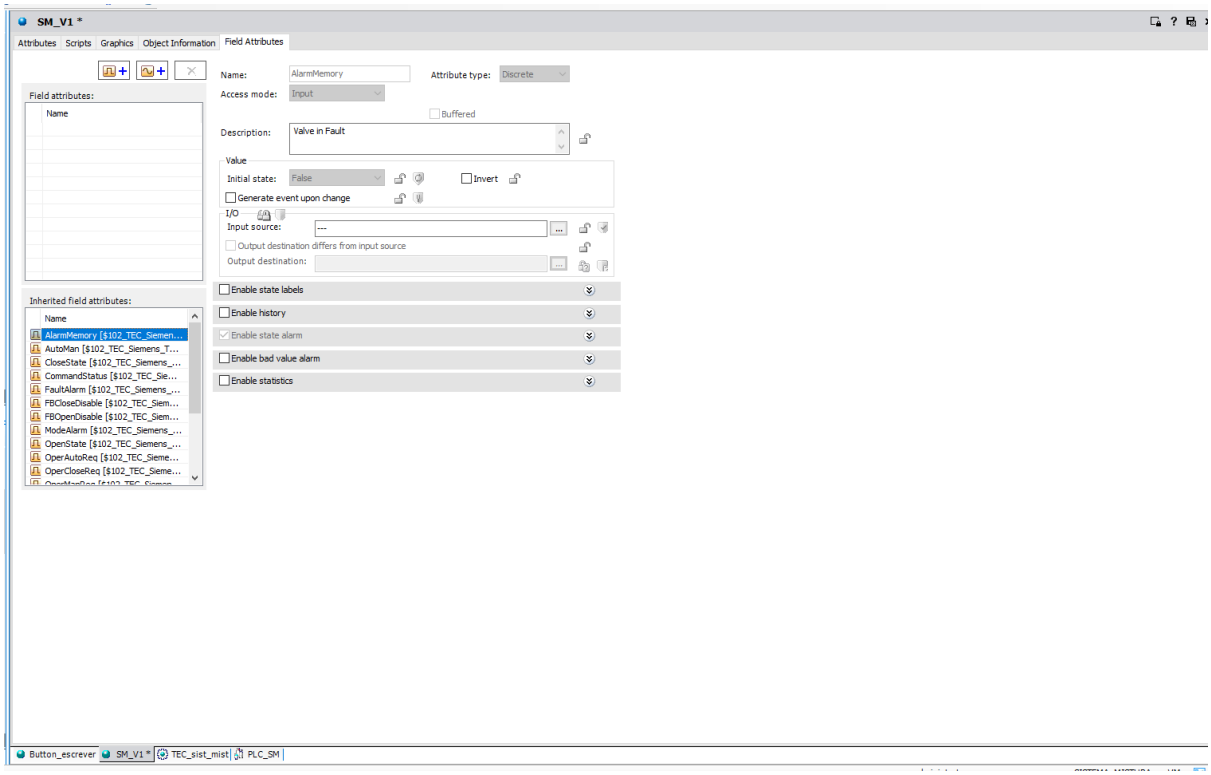


Figura 4.44: Parâmetros de uma válvula

Alguns parâmetros, como por exemplo, o Alarmmemory é selecionado a opção *state alarm*, para estes poderem ser registados como alarmes no histórico de alarmes.

No separador *Attributes* são definidos alguns parâmetros gerais, que vão ser predominantemente utilizados em todos os objetos criados, servindo para auxiliar na comunicação dos vários parâmetros que se encontram presentes no separador *Field Atributtes* (Figura 4.45).

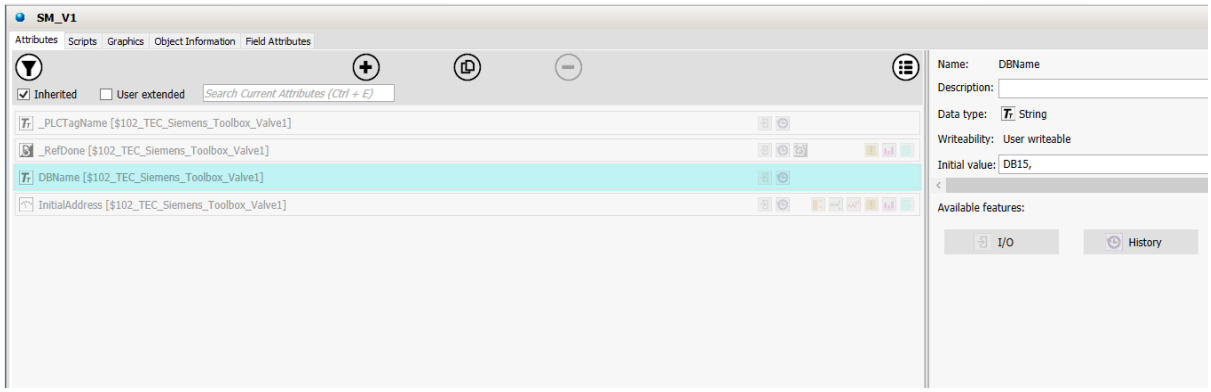


Figura 4.45: Atributos Gerais

Nomeadamente:

- DBName: é inserido o valor do DB a que corresponde o equipamento.
- InitialAddress: corresponde ao primeiro endereço do DB.
- `_RefDone`: trigger para a leitura do script.
- `_PLCTagName`: variável.

No separador scripts é estabelecido o caminho para a comunicação com as variáveis do PLC, como se refere a Figura 4.46.

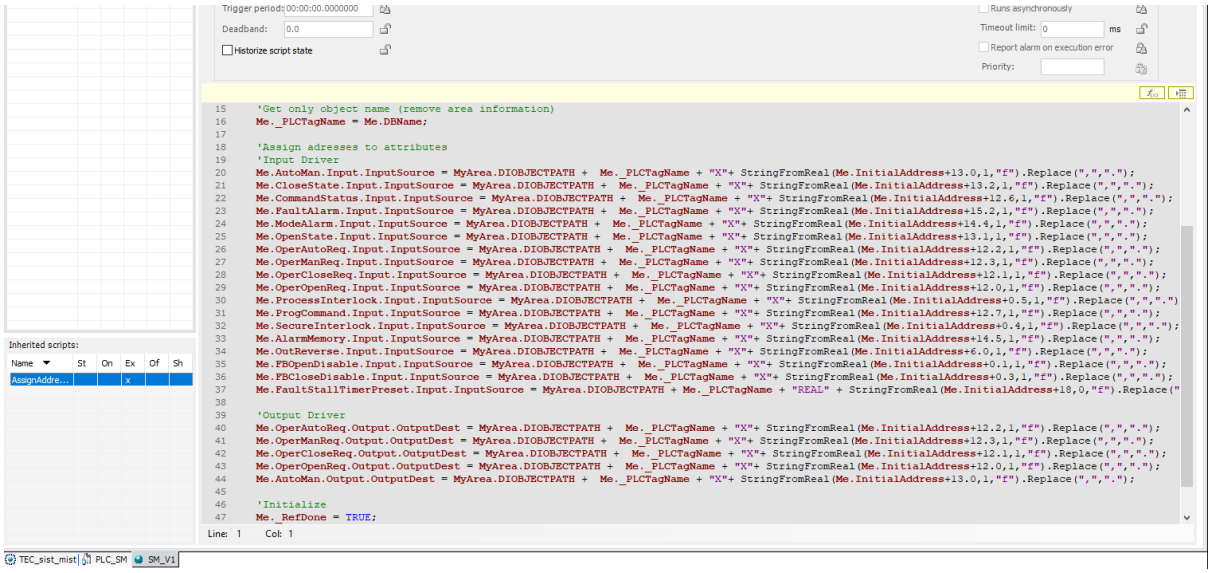


Figura 4.46: Script correspondente a uma válvula

O caminho para cada parâmetro é definido por:

- Diobjectpath: Atributo correspndente ao DDE.
- `_PLCTagName`: Nome do DB a que corresponde o equipamento.
- "X": O atributo é um valor booleano.
- StringFromReal: função que converte de real para string, em que recebe como parâmetros o valor do endereço inicial, o número de casas decimais, a notação exponencial (=1 notação de ponto/ =e exponencial);
- Replace: separação por vírgulas e ponto.

Após o endereçamento de todos os parâmetros, para podermos fazer o debug e vermos se a conexão com as variáveis do PLC estão OK, basta utilizarmos a opção ObjectViewer do objeto, como mostra a Figura 4.47.

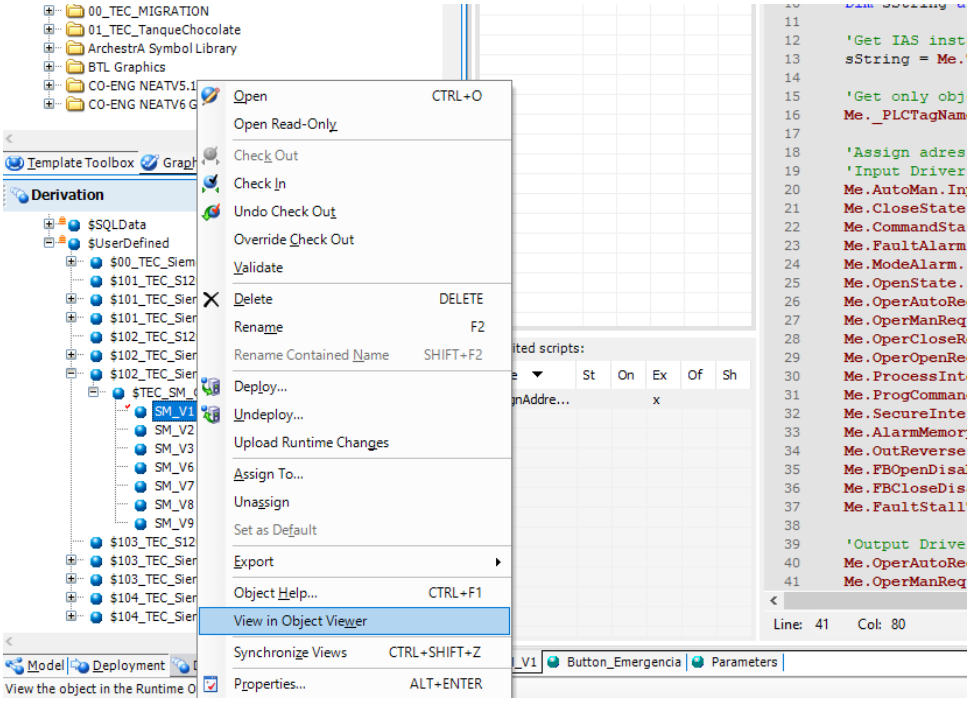


Figura 4.47: Objectviewer

Com a utilização desta ferramenta podemos ver o resultado da nossa conexão e se encontra-se a comunicar com o PLC (Figura 4.48).

A função LogViewer do SMC (Figura 4.49) trata-se de uma ferramenta que permite visualizar e analisar logs gerados pelo sistema. O ArchestrA fornece recursos para o registo de eventos e informações relacionadas à operação do sistema, e o LogViewer é uma ferramenta que ajuda os utilizadores a examinar esses registos para diagnóstico e resolução de problemas.

Contém:

- Registo de Eventos: O ArchestrA pode registar eventos, alarmes, mensagens de

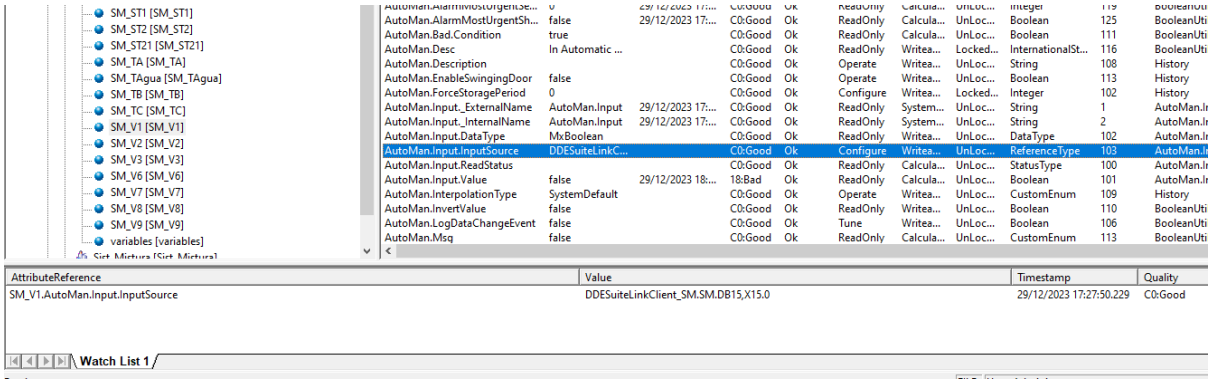


Figura 4.48: Comunicação das variáveis com o PLC

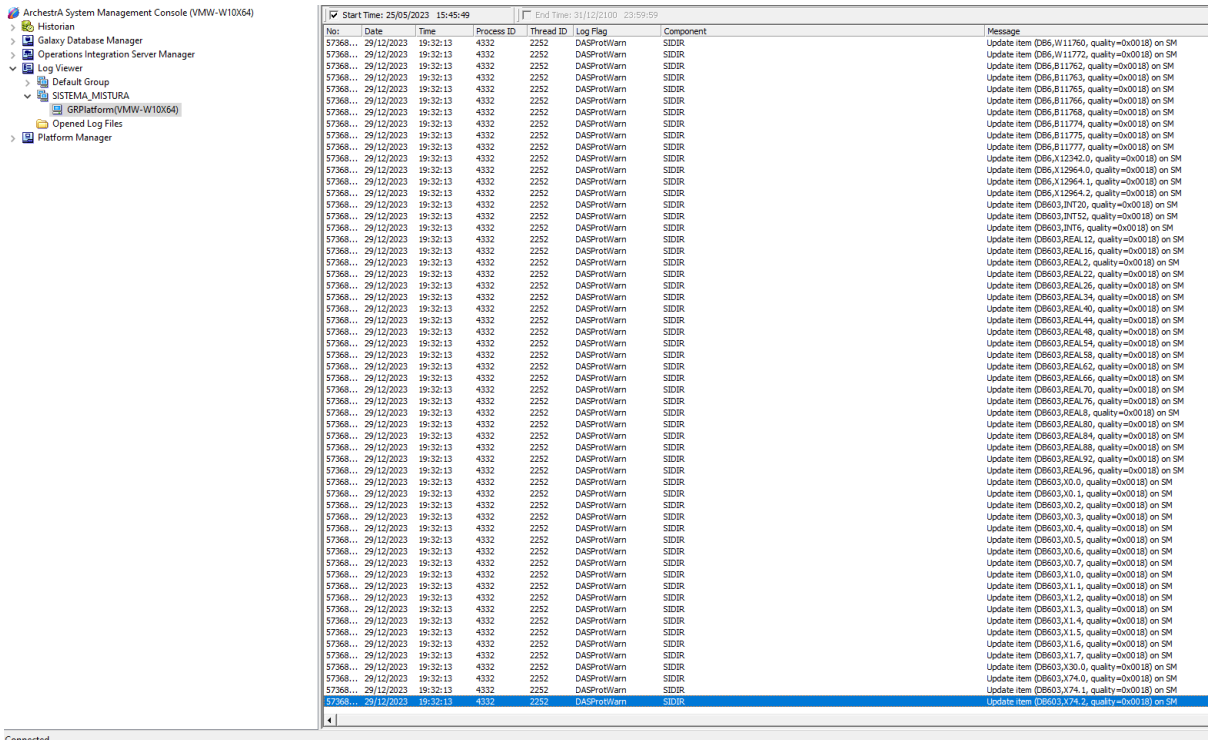


Figura 4.49: LogViewer

sistema e outras informações relevantes durante a operação do sistema. Esses registos são armazenados em arquivos de log.

- Ferramenta de Diagnóstico: O LogViewer serve como uma ferramenta de diagnóstico que permite aos utilizadores examinar e analisar os registos de eventos para entender melhor o comportamento do sistema.
- Filtragem e Pesquisa: O LogViewer geralmente oferece recursos de filtragem e pesquisa para ajudar os utilizadores a localizar eventos específicos ou padrões nos registos.
- Visualização Gráfica: Pode fornecer uma visualização gráfica dos eventos registados, facilitando a identificação de padrões ou anomalias.
- Histórico de Mensagens: O LogViewer pode manter um histórico de mensagens, permitindo aos utilizadores revisar eventos passados e monitorizar a evolução do sistema ao longo do tempo.

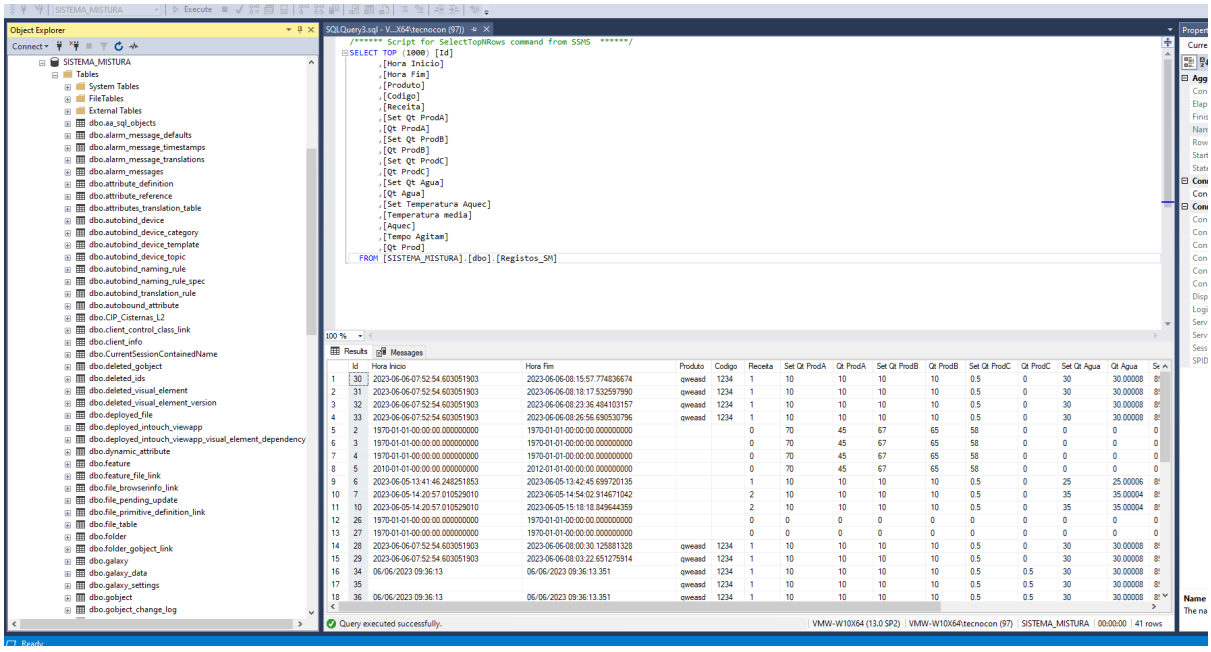


Figura 4.50: Visualização dos registos de receitas no SQL

Para a criação desta tabela de registo (Figura 4.50), foi necessário a criação de um objeto, onde numa primeira fase ele cria a tabela caso não exista e numa segunda fase efetua o registo dos dados. O código desenvolvido para as 2 etapas encontram-se nos anexos C e D.

Nesta versão 2014 do ArchestrA, a criação de uma tabela de alarmes já é feita predefinidamente através da aplicação, em que essa tabela chama-se A2ALMDB. Sendo assim basta simplesmente conectarmos à tabela para podermos visualizar os eventos (Figura 4.51).

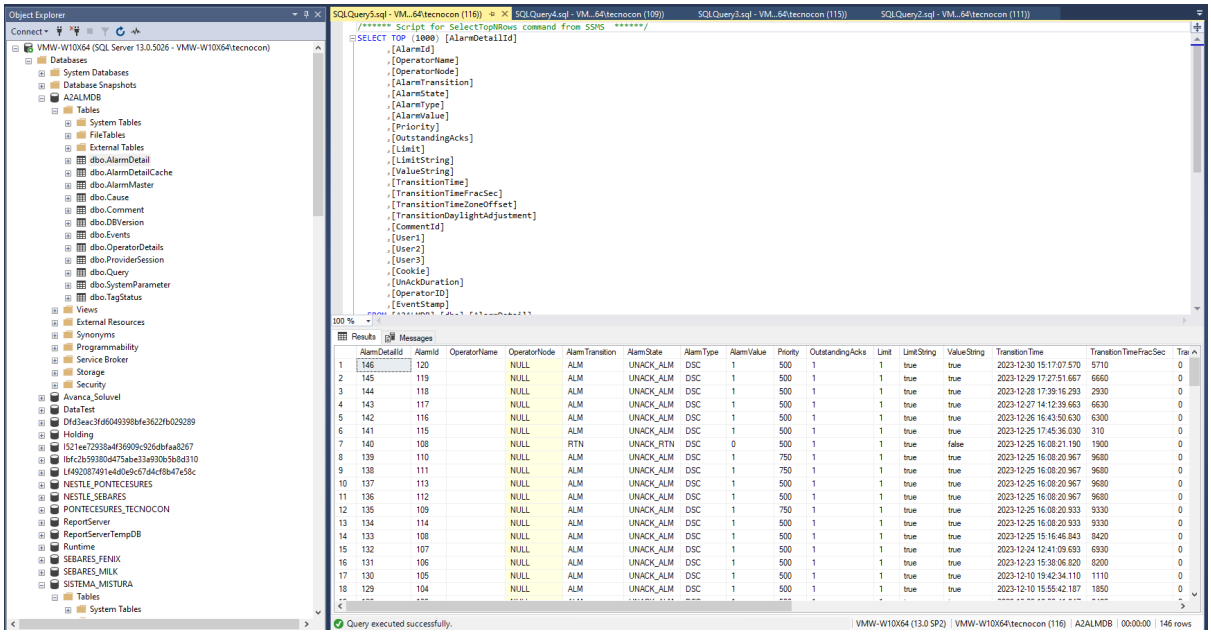


Figura 4.51: Visualização dos registos de alarmes no SQL

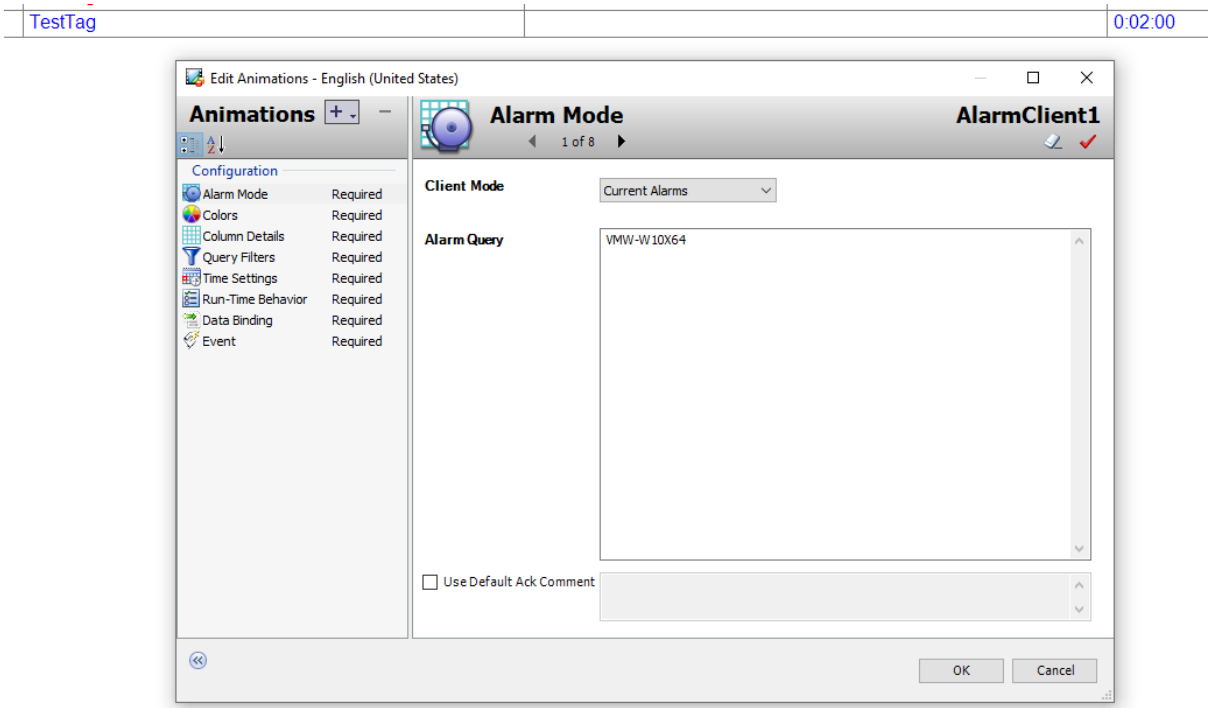


Figura 4.52: Configuração Alarm Query

Pela Figura 4.52, através do objeto que faz o display dos alarmes, só é necessário inserir o nome do SQL server para podermos comunicar.

4.1.4 Conclusão

O presente capítulo apresentou os detalhes técnicos possíveis sobre a aplicação desenvolvida em WinCC e Wonderware. Parte dos detalhes mais específicos não são ilustrados devido para proteger a empresa e o estagiário.

5 Conclusões

Após encerrar o período de estágio, o autor gostaria de expressar a sua gratidão pela oportunidade de contribuir para a Tecnocon, S.A.. Este período foi incrivelmente enriquecedor e proporcionou uma visão valiosa sobre as complexidades e inovações presentes no campo da Automação Industrial com destaque para o *Interface* Homem-Máquina.

Durante os últimos meses, foi possível imergir num projeto desafiador que permitiu aplicar e expandir os conhecimentos em automação industrial e, especificamente, em HMI, autómatos e sistemas SCADA. Participar ativamente no desenvolvimento de *interfaces* intuitivos e eficazes, assim como colaborar com uma equipe tão dedicada, foi uma experiência fundamental para o crescimento profissional.

Ao trabalhar em Siemens (Tia Portal, WinCC) e Wonderware (InTouch), o autor desenvolveu uma compreensão mais profunda de aspetos práticos e teóricos relacionados à integração de consolas HMI em ambientes industriais. Isso incluiu a otimização de processos, garantindo não apenas a eficiência operacional, mas também a segurança e a facilidade de uso para os operadores - foi necessário elaborar o comportamento no autómato e a visualização e parametrição pelos operadores, usando localmente as consolas ou centralmente o sistema SCADA.

Como referido anteriormente, o objetivo desta aplicação é permitir ao seu operador configurar os vários equipamentos necessários para o processo, possibilitar a operação destes mesmos equipamentos em vários modos (manual e automático) e ainda permitir também de forma intuitiva, informativa e funcional de analisar estes mesmos dados.

Durante o período do estágio curricular foi feito o planeamento da aplicação e foi iniciado o seu processo de desenvolvimento. Além disso, a aplicação desenvolvida tem como finalidade corresponder às necessidades de futuros clientes. Essas necessidades podem levar a futuras alterações na aplicação.

Após estas alterações, estarem concluídas e a aplicação seja testada de modo a garantir a sua fiabilidade, terá como finalidade a venda a clientes. Neste tipo de aplicações, a TECNOCON passa por um processo de acompanhamento e suporte às aplicações desenvolvidas. Durante esse período de testes ou mesmo de suporte, pode ser necessário efetuar alterações de modo a corrigir eventuais bugs, ou até adicionar novas funcionalidades que sejam proveitosas.

Em termos de trabalho futuros, propõe-se a exploração mais detalhada das ferramentas

disponibilizadas pelos Wonderware, pois contêm capacidades que permitem automatizar fábricas por completo, usando por exemplo as várias ferramentas como: *System Platform*, *Historian*, *Alarm Adviser*, *Device Integration*, etc.

Referências bibliográficas

[1] Siemens, "Plc s71200," https://www.indiamart.com/proddetail/s71200-siemens-plc-controller-8767840548.html.

[2] Schneider, *Application Server User's Guide*. U.S.A: Schneider, 2015.

[3] Tecnocon, "Tecnocon," https://www.tecnocon.pt.

[4] Aveva, "Wonderware," https://www.aveva.com/en/solutions/operations/wonderware/.

[5] OMRON, "Autómatos programáveis," https://paginas.fe.up.pt/~asousa/tsca/Omron/cursos_omr/Teoria1+2+3_V1_0.pdf, 2015, accessed: 2023-08-06.

[6] A. Parr, *Computers and industrial control*. New York, NY: Industrial Press Inc., 1998.

[7] International Electrotechnical Commission, "Iec 61131-3," https://webstore.iec.ch/publication/4552, 2013.

[8] Siemens, "Wincc," https://www.siemens.com/br/pt/produtos/automacao/simatic-hmi/wincc-unified.html.

[9] Aveva, "Intouch," https://www.aveva.com/en/products/intouch-hmi/.

[10] Schneider, "Schneider," https://www.se.com/pt/pt/product-range/1500-aveva-plant-scada/#overview.

[11] Trihedral, "Vtscada," https://www.vtscada.com/.

[12] Rockwell Automation, "Factorytalk view," https://www.aveva.com/en/solutions/operations/wonderware/.

[13] Siemens, *SIMATIC S7 Programming 1*. German: Siemens, 2003.

[14] Siemens, *TIA Programming Intermediate Skills*. German: Siemens, 2006.

[15] Siemens, "Wincc v7.2 simatic hmi wincc v7.2 getting started," German, 2013.

[16] Siemens, "Operating and monitoring with wincc," German, 2003.

[17] ——, "Tia portal," https://www.siemens.com/global/en/products/automation/industry-software/automation-software/tia-portal.html.

Anexos

Anexo A - Código de leitura de um ficheiro .CSV

Anexo B - Código de escrita para um ficheiro .CSV

Anexo C - Código para criação de tabelas para registos no SQL

Anexo D - Código para escrita de registos no SQL

Anexo A - Código de leitura de um ficheiro .CSV

```
1
2  Sub read4()
3
4  Dim dqNumber, dqNumber2
5  Dim fso, ts, i, D, dteWait
6  Dim start, wait, stdate
7  Dim splitdata, startDate, endDate
8  Dim strSource, strLine, strAll, arrData, strLine1, strLine2
9  Const ForReading = 1
10
11 If SmartTags("read")= True Then
12
13 Set fso = CreateObject("Scripting.FileSystemObject")
14
15   strSource = "C:\Storage Card SD\TEST\ExcelExample\excel_file2.csv"
16
17   SmartTags("concluido") = False
18
19   Set ts = fso.OpenTextFile(strSource,ForReading,0,-2)
20
21     For i=1 To SmartTags("register")
22
23       strLine=ts.ReadLine
24
25       splitdata = Split(strLine, ";")
26
27       SmartTags("find") = True
28
29       If i=2 Then
30
31         'SmartTags("start") = splitdata(0)
32
33         'SmartTags("end") = splitdata(1)
34
35         If DateDiff("h",SmartTags("par_date_inicial"),splitdata(0)) > 0
     Then
36
37           If DateDiff("h",SmartTags("par_date_final"),splitdata(1)) < 0
     Then
38
39             SmartTags("linha") = SmartTags("linha") + 1
40
41             SmartTags("hora_inicial_CSV") = splitdata(0)
42
43             SmartTags("hora_final_CSV") = splitdata(1)
44
```

```
            SmartTags("produto_CSV") = splitdata(2)

            SmartTags("codigo_CSV") = splitdata(3)

            SmartTags("receita_CSV") = splitdata(4)

              SmartTags("set_quant_prodA_CSV") = splitdata(5)

              SmartTags("quant_prodA_CSV") = splitdata(6)

              SmartTags("set_quant_prodB_CSV") = splitdata(7)

            SmartTags("quant_prodB_CSV") = splitdata(8)

              SmartTags("set_quant_prodC_CSV") = splitdata(9)

              SmartTags("quant_prodC_CSV") = splitdata(10)

              SmartTags("set_quant_agua_CSV") = splitdata(11)

            SmartTags("quant_agua_CSV") = splitdata(12)

            SmartTags("set_temp_aquec_CSV") = splitdata(13)

              SmartTags("temp_aquec_CSV") = splitdata(14)

              SmartTags("aquec_CSV") = CBool(splitdata(15))

              SmartTags("temp_agitam_CSV") = splitdata(16)

              SmartTags("quant_prod_CSV") = splitdata(17)

            SmartTags("escreve") = 1

          End If

        End If

      End If

'     dteWait = DateAdd("s", 0.1, Now())
'     Do Until (Now() > dteWait)
'     Loop

      SmartTags("escreve") = 0

      If i=3 Then

```

```
93          'SmartTags("start") = splitdata(0)
94
95          'SmartTags("end") = splitdata(1)
96
97          If DateDiff("h",SmartTags("par_date_inicial"),splitdata(0)) > 0
    Then
98
99            If DateDiff("h",SmartTags("par_date_final"),splitdata(1)) < 0
    Then
100
101             SmartTags("linha") = SmartTags("linha") + 1
102
103             SmartTags("hora_inicial_CSV") = splitdata(0)
104
105             SmartTags("hora_final_CSV") = splitdata(1)
106
107             SmartTags("produto_CSV") = splitdata(2)
108
109             SmartTags("codigo_CSV") = splitdata(3)
110
111             SmartTags("receita_CSV") = splitdata(4)
112
113               SmartTags("set_quant_prodA_CSV") = splitdata(5)
114
115               SmartTags("quant_prodA_CSV") = splitdata(6)
116
117               SmartTags("set_quant_prodB_CSV") = splitdata(7)
118
119             SmartTags("quant_prodB_CSV") = splitdata(8)
120
121               SmartTags("set_quant_prodC_CSV") = splitdata(9)
122
123               SmartTags("quant_prodC_CSV") = splitdata(10)
124
125               SmartTags("set_quant_agua_CSV") = splitdata(11)
126
127               SmartTags("quant_agua_CSV") = splitdata(12)
128
129             SmartTags("set_temp_aquec_CSV") = splitdata(13)
130
131               SmartTags("temp_aquec_CSV") = splitdata(14)
132
133               SmartTags("aquec_CSV") = CBool(splitdata(15))
134
135               SmartTags("temp_agitam_CSV") = splitdata(16)
136
137               SmartTags("quant_prod_CSV") = splitdata(17)
138
```

```
139             SmartTags("escreve") = 1
140
141           End If
142
143         End If
144
145       End If
146
147 '     dteWait = DateAdd("s", 0.1, Now())
148 '     Do Until (Now() > dteWait)
149 '     Loop
150
151       SmartTags("escreve") = 0
152
153       If i=4 Then
154
155         'SmartTags("start") = splitdata(0)
156
157         'SmartTags("end") = splitdata(1)
158
159         If DateDiff("h",SmartTags("par_date_inicial"),splitdata(0)) > 0
    Then
160
161           If DateDiff("h",SmartTags("par_date_final"),splitdata(1)) < 0
    Then
162
163             SmartTags("linha") = SmartTags("linha") + 1
164
165             SmartTags("hora_inicial_CSV") = splitdata(0)
166
167             SmartTags("hora_final_CSV") = splitdata(1)
168
169             SmartTags("produto_CSV") = splitdata(2)
170
171             SmartTags("codigo_CSV") = splitdata(3)
172
173             SmartTags("receita_CSV") = splitdata(4)
174
175               SmartTags("set_quant_prodA_CSV") = splitdata(5)
176
177               SmartTags("quant_prodA_CSV") = splitdata(6)
178
179               SmartTags("set_quant_prodB_CSV") = splitdata(7)
180
181               SmartTags("quant_prodB_CSV") = splitdata(8)
182
183               SmartTags("set_quant_prodC_CSV") = splitdata(9)
184
```

```
185             SmartTags("quant_prodC_CSV") = splitdata(10)
186
187             SmartTags("set_quant_agua_CSV") = splitdata(11)
188
189             SmartTags("quant_agua_CSV") = splitdata(12)
190
191           SmartTags("set_temp_aquec_CSV") = splitdata(13)
192
193             SmartTags("temp_aquec_CSV") = splitdata(14)
194
195             SmartTags("aquec_CSV") = CBool(splitdata(15))
196
197             SmartTags("temp_agitam_CSV") = splitdata(16)
198
199             SmartTags("quant_prod_CSV") = splitdata(17)
200
201           SmartTags("escreve") = 1
202
203         End If
204
205       End If
206
207     End If
208
209     dteWait = DateAdd("s", 0.1, Now())
210     Do Until (Now() > dteWait)
211     Loop
212
213     SmartTags("escreve") = 0
214
215     If i=5 Then
216
217       'SmartTags("start") = splitdata(0)
218
219       'SmartTags("end") = splitdata(1)
220
221       If DateDiff("h",SmartTags("par_date_inicial"),splitdata(0)) > 0
    Then
222
223         If DateDiff("h",SmartTags("par_date_final"),splitdata(1)) < 0
    Then
224
225           SmartTags("linha") = SmartTags("linha") + 1
226
227           SmartTags("hora_inicial_CSV") = splitdata(0)
228
229           SmartTags("hora_final_CSV") = splitdata(1)
230
```

```
            SmartTags("produto_CSV") = splitdata(2)

            SmartTags("codigo_CSV") = splitdata(3)

            SmartTags("receita_CSV") = splitdata(4)

              SmartTags("set_quant_prodA_CSV") = splitdata(5)

              SmartTags("quant_prodA_CSV") = splitdata(6)

              SmartTags("set_quant_prodB_CSV") = splitdata(7)

              SmartTags("quant_prodB_CSV") = splitdata(8)

              SmartTags("set_quant_prodC_CSV") = splitdata(9)

              SmartTags("quant_prodC_CSV") = splitdata(10)

              SmartTags("set_quant_agua_CSV") = splitdata(11)

              SmartTags("quant_agua_CSV") = splitdata(12)
'
            SmartTags("set_temp_aquec_CSV") = splitdata(13)

              SmartTags("temp_aquec_CSV") = splitdata(14)

              SmartTags("aquec_CSV") = CBool(splitdata(15))

              SmartTags("temp_agitam_CSV") = splitdata(16)

              SmartTags("quant_prod_CSV") = splitdata(17)

            SmartTags("escreve") = 1

          End If

        End If

      End If

'     dteWait = DateAdd("s", 0.1, Now())
'     Do Until (Now() > dteWait)
'     Loop

      SmartTags("escreve") = 0

      If i=6 Then

```

```
279         'SmartTags("start") = splitdata(0)
280
281         'SmartTags("end") = splitdata(1)
282
283         If DateDiff("h",SmartTags("par_date_inicial"),splitdata(0)) > 0
    Then
284
285           If DateDiff("h",SmartTags("par_date_final"),splitdata(1)) < 0
    Then
286
287             SmartTags("linha") = SmartTags("linha") + 1
288
289             SmartTags("hora_inicial_CSV") = splitdata(0)
290
291             SmartTags("hora_final_CSV") = splitdata(1)
292
293             SmartTags("produto_CSV") = splitdata(2)
294
295             SmartTags("codigo_CSV") = splitdata(3)
296
297             SmartTags("receita_CSV") = splitdata(4)
298
299               SmartTags("set_quant_prodA_CSV") = splitdata(5)
300
301               SmartTags("quant_prodA_CSV") = splitdata(6)
302
303               SmartTags("set_quant_prodB_CSV") = splitdata(7)
304
305               SmartTags("quant_prodB_CSV") = splitdata(8)
306
307               SmartTags("set_quant_prodC_CSV") = splitdata(9)
308
309               SmartTags("quant_prodC_CSV") = splitdata(10)
310
311               SmartTags("set_quant_agua_CSV") = splitdata(11)
312
313               SmartTags("quant_agua_CSV") = splitdata(12)
314
315             SmartTags("set_temp_aquec_CSV") = splitdata(13)
316
317               SmartTags("temp_aquec_CSV") = splitdata(14)
318
319               SmartTags("aquec_CSV") = CBool(splitdata(15))
320
321               SmartTags("temp_agitam_CSV") = splitdata(16)
322
323               SmartTags("quant_prod_CSV") = splitdata(17)
324
```

```
325             SmartTags("escreve") = 1
326
327           End If
328
329         End If
330
331       End If
332
333 '     dteWait = DateAdd("s", 0.1, Now())
334 '     Do Until (Now() > dteWait)
335 '     Loop
336
337       SmartTags("escreve") = 0
338
339       If i=7 Then
340
341         'SmartTags("start") = splitdata(0)
342
343         'SmartTags("end") = splitdata(1)
344
345         If DateDiff("h",SmartTags("par_date_inicial"),splitdata(0)) > 0
    Then
346
347           If DateDiff("h",SmartTags("par_date_final"),splitdata(1)) < 0
    Then
348
349             SmartTags("linha") = SmartTags("linha") + 1
350
351             SmartTags("hora_inicial_CSV") = splitdata(0)
352
353             SmartTags("hora_final_CSV") = splitdata(1)
354
355             SmartTags("produto_CSV") = splitdata(2)
356
357             SmartTags("codigo_CSV") = splitdata(3)
358
359             SmartTags("receita_CSV") = splitdata(4)
360
361               SmartTags("set_quant_prodA_CSV") = splitdata(5)
362
363               SmartTags("quant_prodA_CSV") = splitdata(6)
364
365               SmartTags("set_quant_prodB_CSV") = splitdata(7)
366
367               SmartTags("quant_prodB_CSV") = splitdata(8)
368
369               SmartTags("set_quant_prodC_CSV") = splitdata(9)
370
```

```
371             SmartTags("quant_prodC_CSV") = splitdata(10)
372
373           SmartTags("set_temp_aquec_CSV") = splitdata(13)
374
375             SmartTags("temp_aquec_CSV") = splitdata(14)
376
377             SmartTags("aquec_CSV") = CBool(splitdata(15))
378
379             SmartTags("temp_agitam_CSV") = splitdata(16)
380
381             SmartTags("quant_prod_CSV") = splitdata(17)
382
383           SmartTags("escreve") = 1
384
385         End If
386
387       End If
388
389     End If
390
391     dteWait = DateAdd("s", 0.1, Now())
392     Do Until (Now() > dteWait)
393     Loop
394
395     SmartTags("escreve") = 0
396
397     If i=8 Then
398
399       'SmartTags("start") = splitdata(0)
400
401       'SmartTags("end") = splitdata(1)
402
403       If DateDiff("h",SmartTags("par_date_inicial"),splitdata(0)) > 0
    Then
404
405         If DateDiff("h",SmartTags("par_date_final"),splitdata(1)) < 0
    Then
406
407           SmartTags("linha") = SmartTags("linha") + 1
408
409           SmartTags("hora_inicial_CSV") = splitdata(0)
410
411           SmartTags("hora_final_CSV") = splitdata(1)
412
413           SmartTags("produto_CSV") = splitdata(2)
414
415           SmartTags("codigo_CSV") = splitdata(3)
416
```

```
417             SmartTags("receita_CSV") = splitdata(4)
418
419               SmartTags("set_quant_prodA_CSV") = splitdata(5)
420
421               SmartTags("quant_prodA_CSV") = splitdata(6)
422
423               SmartTags("set_quant_prodB_CSV") = splitdata(7)
424
425             SmartTags("quant_prodB_CSV") = splitdata(8)
426
427               SmartTags("set_quant_prodC_CSV") = splitdata(9)
428
429               SmartTags("quant_prodC_CSV") = splitdata(10)
430
431             SmartTags("set_temp_aquec_CSV") = splitdata(13)
432
433               SmartTags("temp_aquec_CSV") = splitdata(14)
434
435               SmartTags("aquec_CSV") = CBool(splitdata(15))
436
437               SmartTags("temp_agitam_CSV") = splitdata(16)
438
439               SmartTags("quant_prod_CSV") = splitdata(17)
440
441             SmartTags("escreve") = 1
442
443           End If
444
445         End If
446
447       End If
448
449 '     dteWait = DateAdd("s", 0.1, Now())
450 '     Do Until (Now() > dteWait)
451 '     Loop
452
453       SmartTags("escreve") = 0
454
455       If i=9 Then
456
457         'SmartTags("start") = splitdata(0)
458
459         'SmartTags("end") = splitdata(1)
460
461         If DateDiff("h",SmartTags("par_date_inicial"),splitdata(0)) > 0
    Then
462
463           If DateDiff("h",SmartTags("par_date_final"),splitdata(1)) < 0
```

```
    Then
464
465             SmartTags("linha") = SmartTags("linha") + 1
466
467             SmartTags("hora_inicial_CSV") = splitdata(0)
468
469             SmartTags("hora_final_CSV") = splitdata(1)
470
471             SmartTags("produto_CSV") = splitdata(2)
472
473             SmartTags("codigo_CSV") = splitdata(3)
474
475             SmartTags("receita_CSV") = splitdata(4)
476
477               SmartTags("set_quant_prodA_CSV") = splitdata(5)
478
479               SmartTags("quant_prodA_CSV") = splitdata(6)
480
481               SmartTags("set_quant_prodB_CSV") = splitdata(7)
482
483               SmartTags("quant_prodB_CSV") = splitdata(8)
484
485               SmartTags("set_quant_prodC_CSV") = splitdata(9)
486
487               SmartTags("quant_prodC_CSV") = splitdata(10)
488
489               SmartTags("set_quant_agua_CSV") = splitdata(11)
490
491               SmartTags("quant_agua_CSV") = splitdata(12)
492
493             SmartTags("set_temp_aquec_CSV") = splitdata(13)
494
495               SmartTags("temp_aquec_CSV") = splitdata(14)
496
497               SmartTags("aquec_CSV") = CBool(splitdata(15))
498
499               SmartTags("temp_agitam_CSV") = splitdata(16)
500
501               SmartTags("quant_prod_CSV") = splitdata(17)
502
503             SmartTags("escreve") = 1
504
505           End If
506
507         End If
508
509       End If
510
```

```
'      dteWait = DateAdd("s", 0.1, Now())
'      Do Until (Now() > dteWait)
'      Loop

       SmartTags("escreve") = 0

       If i=10 Then

         'SmartTags("start") = splitdata(0)

         'SmartTags("end") = splitdata(1)

         If DateDiff("h",SmartTags("par_date_inicial"),splitdata(0)) > 0
    Then

           If DateDiff("h",SmartTags("par_date_final"),splitdata(1)) < 0
    Then

             SmartTags("linha") = SmartTags("linha") + 1

             SmartTags("hora_inicial_CSV") = splitdata(0)

             SmartTags("hora_final_CSV") = splitdata(1)

             SmartTags("produto_CSV") = splitdata(2)

             SmartTags("codigo_CSV") = splitdata(3)

             SmartTags("receita_CSV") = splitdata(4)

               SmartTags("set_quant_prodA_CSV") = splitdata(5)

               SmartTags("quant_prodA_CSV") = splitdata(6)

               SmartTags("set_quant_prodB_CSV") = splitdata(7)

               SmartTags("quant_prodB_CSV") = splitdata(8)

               SmartTags("set_quant_prodC_CSV") = splitdata(9)

               SmartTags("quant_prodC_CSV") = splitdata(10)

               SmartTags("set_quant_agua_CSV") = splitdata(11)

               SmartTags("quant_agua_CSV") = splitdata(12)

             SmartTags("set_temp_aquec_CSV") = splitdata(13)

```

```
557             SmartTags("temp_aquec_CSV") = splitdata(14)
558
559             SmartTags("aquec_CSV") = CBool(splitdata(15))
560
561             SmartTags("temp_agitam_CSV") = splitdata(16)
562
563             SmartTags("quant_prod_CSV") = splitdata(17)
564
565           SmartTags("escreve") = 1
566
567         End If
568
569       End If
570
571     End If
572
573     dteWait = DateAdd("s", 0.1, Now())
574     Do Until (Now() > dteWait)
575     Loop
576
577     SmartTags("escreve") = 0
578
579     If i=11 Then
580
581       'SmartTags("start") = splitdata(0)
582
583       'SmartTags("end") = splitdata(1)
584
585       If DateDiff("h",SmartTags("par_date_inicial"),splitdata(0)) > 0
    Then
586
587         If DateDiff("h",SmartTags("par_date_final"),splitdata(1)) < 0
    Then
588
589           SmartTags("linha") = SmartTags("linha") + 1
590
591           SmartTags("hora_inicial_CSV") = splitdata(0)
592
593           SmartTags("hora_final_CSV") = splitdata(1)
594
595           SmartTags("produto_CSV") = splitdata(2)
596
597           SmartTags("codigo_CSV") = splitdata(3)
598
599           SmartTags("receita_CSV") = splitdata(4)
600
601             SmartTags("set_quant_prodA_CSV") = splitdata(5)
602
```

```
603                 SmartTags("quant_prodA_CSV") = splitdata(6)
604
605                 SmartTags("set_quant_prodB_CSV") = splitdata(7)
606
607                 SmartTags("quant_prodB_CSV") = splitdata(8)
608
609                 SmartTags("set_quant_prodC_CSV") = splitdata(9)
610
611                 SmartTags("quant_prodC_CSV") = splitdata(10)
612
613                 SmartTags("set_quant_agua_CSV") = splitdata(11)
614
615                 SmartTags("quant_agua_CSV") = splitdata(12)
616
617               SmartTags("set_temp_aquec_CSV") = splitdata(13)
618
619                 SmartTags("temp_aquec_CSV") = splitdata(14)
620
621                 SmartTags("aquec_CSV") = CBool(splitdata(15))
622
623                 SmartTags("temp_agitam_CSV") = splitdata(16)
624
625                 SmartTags("quant_prod_CSV") = splitdata(17)
626
627               SmartTags("escreve") = 1
628
629             End If
630
631           End If
632
633         End If
634
635 '       dteWait = DateAdd("s", 0.1, Now())
636 '       Do Until (Now() > dteWait)
637 '       Loop
638
639         SmartTags("escreve") = 0
640
641         If i=12 Then
642
643           'SmartTags("start") = splitdata(0)
644
645           'SmartTags("end") = splitdata(1)
646
647           If DateDiff("h",SmartTags("par_date_inicial"),splitdata(0)) > 0
      Then
648
649             If DateDiff("h",SmartTags("par_date_final"),splitdata(1)) < 0
```

```
    Then
650
651             SmartTags("linha") = SmartTags("linha") + 1
652
653             SmartTags("hora_inicial_CSV") = splitdata(0)
654
655             SmartTags("hora_final_CSV") = splitdata(1)
656
657             SmartTags("produto_CSV") = splitdata(2)
658
659             SmartTags("codigo_CSV") = splitdata(3)
660
661             SmartTags("receita_CSV") = splitdata(4)
662
663               SmartTags("set_quant_prodA_CSV") = splitdata(5)
664
665             SmartTags("quant_prodA_CSV") = splitdata(6)
666
667               SmartTags("set_quant_prodB_CSV") = splitdata(7)
668
669               SmartTags("quant_prodB_CSV") = splitdata(8)
670
671               SmartTags("set_quant_prodC_CSV") = splitdata(9)
672
673               SmartTags("quant_prodC_CSV") = splitdata(10)
674
675               SmartTags("set_quant_agua_CSV") = splitdata(11)
676
677               SmartTags("quant_agua_CSV") = splitdata(12)
678
679             SmartTags("set_temp_aquec_CSV") = splitdata(13)
680
681               SmartTags("temp_aquec_CSV") = splitdata(14)
682
683               SmartTags("aquec_CSV") = CBool(splitdata(15))
684
685               SmartTags("temp_agitam_CSV") = splitdata(16)
686
687               SmartTags("quant_prod_CSV") = splitdata(17)
688
689             SmartTags("escreve") = 1
690
691           End If
692
693         End If
694
695       End If
696
```

```
697
698       If i=SmartTags("register") Then
699
700         SmartTags("procurar") = False
701
702         SmartTags("find") = False
703
704         SmartTags("concluido")= True
705
706       End If
707
708     Next
709
710 End If
711
712 ts.Close
713
714 Set ts  = Nothing
715
716 Set fso = Nothing
717
718 HmiRuntime.SmartTags("read")= False
719
720 End Sub
```

Listing 5.1: leitura de um ficheiro .CSV

Anexo B - Código de escrita para um ficheiro .CSV

```
1
2  Sub Write_data()
3
4  Dim Folderway, ObjectWay, FileName, File, FileExist, Apendix, Row, i
5
6  If HmiRuntime.SmartTags("write") = True Then
7
8
9    Folderway = "C:\Storage Card SD\TEST\ExcelExample"
10
11   Set ObjectWay = CreateObject("Scripting.FileSystemObject")
12
13
14   If Not ObjectWay.FolderExists(Folderway) Then
15
16     ObjectWay.CreateFolder Folderway
17
18   End If
19
20
21   FileName = "excel_file2.csv"
22
23   Set File = CreateObject("Scripting.FileSystemObject")
24
25   FileExist = File.FileExists(Folderway & "\" & FileName)
26
27
28   If FileExist = False Then
29
30     File.CreateTextFile(Folderway &"\"& FileName)
31
32     Set Apendix = File.OpenTextFile(Folderway &"\"& FileName,8)
33
34     Apendix.WriteLine(" Hora Inicial ; Hora Final ; Produto ; C digo ;
       Receita ; Set Qt. Prod.A ; Qt. Prod.A ; Set Qt. Prod.B ; Qt. Prod.B
       ; Set Qt. Prod.C ; Qt. Prod.C ; Set Qt.  gua  ; Qt.  gua  ; Set
     Temperatura Aquec. ; Temperatura M dia ; Aquec. ; Tempo Agitam. ;
     Qt. Produzida ;")
35
36     Apendix.Close
37
38     Set File = Nothing
39
40   End If
41
42
```

```
43    Set File = CreateObject("Scripting.FileSystemObject")
44
45    Set Row = File.OpenTextFile(Folderway &"\"& FileName,8)
46
47    Row.WriteLine(SmartTags("hora_inicial") & ";" &  SmartTags("
       hora_final") & ";" & SmartTags("produto") & ";" & SmartTags("codigo"
       ) & ";" & SmartTags("receita") & ";" & SmartTags("set_quant_prodA")
       & ";" & SmartTags("quant_prodA") & ";" & SmartTags("set_quant_prodB"
       ) & ";" & SmartTags("quant_prodB") & ";" & SmartTags("
       set_quant_prodC") & ";" & SmartTags("quant_prodC") & ";" & SmartTags
       ("set_quant_agua") & ";" & SmartTags("quant_agua") & ";" & SmartTags
       ("set_temp_aquec") & ";" & SmartTags("temp_aquec") & ";" & SmartTags
       ("aquec") & ";" & SmartTags("tempo_agitam") & ";" &SmartTags("
       quant_produzida"))
48
49    Row.Close
50
51  End If
52
53  HmiRuntime.SmartTags("write") = False
54
55  End Sub
```

Listing 5.2: escrita num ficheiro .CSV

Anexo C - Código para criação de tabelas para registos no SQL

```
1
2  '#######################################################
3  '###          VALIDATE TABLES
4  '#######################################################
5
6  DIM str as string;
7  me.BtnValidateTables=false;
8
9  DIM TecSqla as TecSQLfunctions.TecSQL;
10 DIM TecFct as TecSQLFunctions.TecFunctions;
11
12 ' *** Valida se existem as tabelas ***
13
14 ' ####
15 str="IF NOT EXISTS (Select * from information_schema.tables where
       table_name = 'Registos_SM10') "+
16 + " Begin create table Registos_SM10 ([Id] [numeric] Identity (1,1) not
       null,[Hora Inicio] [nvarchar] (max))"+
17 + " END";
18 logmessage(str);
19
20 if TecSqla.ExecutarQuery(me.StrCon, str ) then
21 logmessage("Tabela Registos_SM Validada.");
22 endif;
```

Listing 5.3: criação de tabelas para registos no SQL

Anexo D - Código para escrita de registos no SQL

```
'#########################################################
'###                  DATA REGISTER                    ###
'#########################################################
'SEQ_S01.Produccion
DIM str as string;

DIM TecSqla as TecSQLfunctions.TecSQL;
DIM TecFct as TecSQLFunctions.TecFunctions;

dim str1 as string;
dim str2 as string;
dim str3 as string;
dim str4 as string;
dim str5 as string;
dim str6 as string;
dim str7 as string;
dim str8 as string;
dim str9 as string;
dim str10 as string;
dim str11 as string;
dim str12 as string;
dim str13 as string;
dim str14 as string;
dim str15 as string;
dim str16 as string;
dim str17 as string;
dim str18 as string;

try

LogMessage("str16 : " + str16);
' #### Insert
str="Insert into [Registos_SM] " +
    "([Hora Inicio]" +
    ",[Hora Fim]" +
    ",[Produto]" +
    ",[Codigo]" +
    ",[Receita]" +
    ",[Set Qt ProdA]" +
    ",[Qt ProdA]" +
  ",[Set Qt ProdB]" +
    ",[Qt ProdB]" +
    ",[Set Qt ProdC]" +
    ",[Qt ProdC]" +
    ",[Set Qt Agua]" +
```

```
47      ",[Qt Agua]" +
48      ",[Set Temperatura Aquec]" +
49      ",[Temperatura media]" +
50      ",[Aquec]" +
51      ",[Tempo Agitam]" +
52      ",[Qt Prod])" +
53    " Values " +
54    " ('"+Me.Hora_Inicio+"','"+Me.Hora_Fim+"','"+Me.Produto+"','"+Me.
      Codigo+"','"+Me.Receita+"','"+Me.Set_Qt_ProdA+"','"+Me.Qt_ProdA+"','
      "+Me.Set_Qt_ProdB+"','"+Me.Qt_ProdB+
55    "','"+Me.Set_Qt_ProdC+"','"+Me.Qt_ProdC+"','"+Me.Set_Qt_Agua+"','"+Me
      .Qt_Agua+"','"+Me.Set_Temp_Aquec+"','"+Me.Temp_media+"','"+Me.Aquec+
      "','"+Me.Temp_Agitam+
56    "','"+Me.Quant_Prod+"')";
57
58
59  if TecSqla.ExecutarQuery(me.StrCon, str ) then
60  logmessage("Save data Registos SM.");
61  logmessage(str);
62  else
63  logmessage("Registos SM Error.");
64  endif;
65
66  catch ' If SCRIPT error
67  logmessage("Registos SM Error: "+error);
68
69  endtry;
70  me.write=false;
```

Listing 5.4: escrita de registos no SQL

Instituto Superior de Engenharia
Politécnico de Coimbra