



XXIV SNPTEE  
SEMINÁRIO NACIONAL DE PRODUÇÃO E  
TRANSMISSÃO DE ENERGIA ELÉTRICA

CB/GPT/04

22 a 25 de outubro de 2017  
Curitiba - PR

## GRUPO - 2

### GRUPO DE ESTUDO DE PRODUÇÃO TÉRMICA E FONTES NÃO CONVENCIONAIS - GPT

#### SISTEMA DE AQUISIÇÃO DE DADOS UTILIZANDO TECNOLOGIA OPC PARA CONSTRUÇÃO DE MODELOS LÓGICO-MATEMÁTICOS DE UMA TURBINA A GÁS HEAVY DUTY

**Luis Gustavo T. Ribas(\*)**  
INSTITUTOS LACTEC

**Eduardo M. Yamao**  
INSTITUTOS LACTEC

**Fabio Y. Nara**  
INSTITUTOS LACTEC

**Felipe T. R. Tovar**  
INSTITUTOS LACTEC

**João Borba**  
INSTITUTOS LACTEC

**Flavio Chiesa**  
UEGA

**Victor M.L. Santos**  
UEGA

**Andre D. S. Orlandi**  
COPEL

## RESUMO

O objetivo deste artigo é apresentar um estudo de caso de uso da tecnologia OPC Classic 2.0 para integrar as informações oriundas de diversos sistemas de controle da Usina Termelétrica a Gás Natural – UEGA em um sistema de aquisição de dados para a construção de modelos lógico-matemáticos de uma turbina a gás *Heavy Duty*. O sistema foi desenvolvido com o intuito de coletar e criar um histórico de dados do processo de combustão da turbina para fins de otimização e simulação de sua operação. Ele foi desenvolvido para operar em conjunto com outros sistemas da turbina como SPPA T3000 da Siemens e o 800xA da ABB, recolhendo dados de diversas fontes. Sua implantação foi realizada com sucesso e os dados já estão sendo utilizados para criação e validação dos modelos da turbina como também para a criação de uma interface com o usuário com informações relevantes sobre a operação da planta.

## PALAVRAS-CHAVE

Sistema de aquisição, Turbinas à gás Heavy Duty, OPC, Modelos matemáticos

## 1.0 - INTRODUÇÃO

O objetivo deste artigo é apresentar um estudo de caso do uso da tecnologia OPC para integrar as informações oriundas de diversos sistemas de controle da Usina Termelétrica a Gás Natural de Araucária – UEGA - visando a construção e simulação de modelos lógico-matemáticos de uma turbina a gás *Heavy Duty*. Os modelos lógico-matemáticos farão parte de um sistema computacional cujo objetivo é minimizar o consumo de combustível, mantendo as pulsações na câmara de combustão e os níveis de emissões dentro de limites estabelecidos de forma a resguardar a integridade das máquinas e respeitar as normas ambientais vigentes. Estes modelos foram construídos, ajustados e validados com base em dados reais da planta a fim de obter uma melhor aproximação do processo real. Para dar suporte a tal sistema, tanto no momento de sua construção quanto a partir do momento em que este seja posto em operação, faz-se necessário disponibilizar continuamente dados de inúmeras variáveis do processo, e armazená-los formando um registro histórico. Estes dados correspondem a leituras dos diversos sensores e atuadores instalados na planta, e conectados a diferentes sistemas de controle, sendo os principais o sistema SPPA T3000 da Siemens e o 800xA da ABB, responsáveis pelo controle e operação de forma conjunta do processo de ciclo combinado. Para realizar a aquisição, armazenamento e disponibilização destes dados ao sistema a ser construído criou-se um programa específico, a ser apresentado no presente artigo.

A solução adotada foi o emprego da tecnologia OPC Classic 2.0 que permitiu a aquisição dos dados do processo das diferentes fontes e consolidando-os em uma base de dados própria a fim de desacoplar os sistemas de controle da usina e o sistema baseado em simulação a ser construído. No total, cerca de cinco mil variáveis foram selecionadas para comporem a base de dados, sendo capturadas a uma taxa de 1 amostra de cada variável por segundo. Os dados obtidos continuamente através das conexões OPC são então armazenadas em um banco de dados relacional, formando uma camada intermediária entre os sistemas de controle da usina e o sistema

(\*) Avenida Prefeito Lothário Meissner, n° 01 – CEP 80210170 Curitiba, PR – Brasil

baseado em simulação. Este banco de dados está estruturado de tal forma que suporte o registro de informações do montante de dados na frequência especificada, além de paralelamente disponibilizar tais dados para diferentes tipos de consulta, seja pelos dados mais recentes de um conjunto de variáveis ou do histórico de valores das variáveis.

O sistema de aquisição e armazenamento de dados resultante foi projetado desta forma também por motivos de segurança da rede industrial da usina, separando o mecanismo de aquisição de dados da base onde os dados adquiridos são registrados em máquinas distintas conectadas pela rede através de um *firewall*. Sendo assim, criou-se um isolamento, no nível do fluxo de dados, entre a rede industrial e a rede da UEGA, onde o sistema de simulação e otimização será instalado.

O sistema de aquisição e armazenamento de dados de processo foi desenvolvido e implantado com sucesso, e os dados coletados vem sendo empregados no desenvolvimento e validação dos modelos lógico-matemáticos que fazem parte do sistema baseado em simulação para o processo de combustão das turbinas a gás da usina, e também será empregado posteriormente diretamente por tal sistema, tanto para disponibilizar visualizações de dados para o usuário, quanto para parametrizar as simulações.

## 2.0 - TECNOLOGIAS EMPREGADAS

O sistema de aquisição e armazenamento de dados foi desenvolvido empregando algumas tecnologias que formam a base funcional do sistema. Mais especificamente, em se tratando da aquisição de dados, as tecnologias de comunicação industrial OPC e Modbus foram utilizadas possibilitando à ferramenta interagir com os sistemas de controle da planta para a obtenção de dados. Quanto ao armazenamento destes, empregou-se sistemas de banco de dados relacionais para realizar a persistência das aquisições em disco de forma organizada e de fácil acesso. Ainda, o desenvolvimento do sistema em si emprega a plataforma LabVIEW para a implementação da aplicação propriamente dita. A seguir, segue-se o detalhamento descritivo destas tecnologias, explanando os conceitos teóricos destas tecnologias e as consequências práticas de suas aplicações no desenvolvimento da ferramenta de aquisição de dados.

### 2.1 – OPC

O acrônimo "OPC" vem originalmente de "*OLE (Object Linking and Embedding) for Process Control*", mas hoje em dia a sigla OPC já não tem um significado oficial. O OPC foi projetado para ser uma camada de abstração entre redes industriais e controladores CLP (Controlador Lógico Programável) proprietários. O padrão OPC possui duas variações, o OPC *Classic* e o OPC UA. Cada uma delas define um conjunto de interfaces não determinísticas que permitem que qualquer cliente possa acessar qualquer dispositivo compatível com o padrão OPC, diminuindo os esforços de duplicação no desenvolvimento de protocolos específicos de dispositivos, eliminando inconsistências entre dispositivos, fornecendo suporte para alterações de recursos de *hardware* e evitando conflitos de acesso em sistemas de controle industrial. Embora a intenção básica de OPC seja fornecer uma camada intermediária abrangente ele é comumente utilizado para acesso direto a dados de processo, ou como uma ponte de comunicação entre sistemas supervisórios (WOLFGANG MAHNKE, 2009).

No OPC não há aplicativos clientes autônomos, já que o objetivo de um cliente OPC é fornecer um ponto de comunicação e de troca de dados para outras aplicações, como sistemas SCADA, DCS ou outras ferramentas. Sendo assim, o OPC torna-se um bloco de construção para muitos sistemas complexos como o MES (Sistema de Execução de Manufatura) e ERP (Planejamento de Recursos Empresariais).

O padrão OPC especifica o comportamento hierárquico das interfaces cliente/servidor, que basicamente giram em torno do modo de acesso a três elementos básicos desta interface, a saber:

- **Item:** Este é o objeto OPC, que contém um identificador exclusivo que atua como uma referência de acesso para a fonte de dados, bem como o valor, *status* e *timestamp*, onde:
  - *Valor:* é o dado da fonte;
  - *Status:* é o estado de qualidade da informações sobre o dispositivo;
  - *Timestamp:* é a hora em que os dados foram captados.
- **Group:** Elemento que agrupa e organiza os itens OPC além de manter informações sobre si, como taxa de atualização, forma de atualização entre outros;
- **Server:** Elemento que agrupa e organiza o *Group*. Este elemento é a referência para acesso ao cliente OPC.

#### 2.1.1 – Obtenção dos dados

A obtenção dos dados de um servidor OPC através de um cliente pode ser realizada de duas formas: modo Síncrono e modo Assíncrono. No modo Síncrono um aplicativo cliente OPC emite uma solicitação síncrona, e então esperará o servidor OPC fornecer o resultado para aquela solicitação. Esta funcionalidade torna mais fácil o desenvolvimento de aplicações que necessitam de uma sequência específica de eventos.

Por exemplo, quando um procedimento operacional exige a abertura de uma válvula antes de ligar uma bomba, pode-se primeiramente executar uma chamada síncrona para abrir a válvula, e então usar uma chamada síncrona para ligar a bomba. O funcionamento sincronizado garante que a primeira ação (abrindo a válvula) seja executada completamente antes da segunda ação (ligar a bomba). Em relação ao uso de rede, a leitura síncrona de um servidor pode ocasionar um uso elevado de banda pois todos os elementos OPC solicitados serão transmitidos, independente se houve ou não houve mudança em seu valor.

Já no modo Assíncrono, quando um aplicativo cliente OPC emite uma chamada assíncrona, o servidor OPC confirma o recebimento da própria chamada. Em seguida, "chama" o aplicativo cliente OPC de volta com o resultado usando um *call-back* quando este estiver disponível. O aplicativo cliente não sabe quando ele irá receber a chamada de retorno, e, portanto, a chamada diz-se assíncrona, ou sem referência de temporização. Utilizam-se chamadas assíncronas quando a aplicação não exige uma ordem específica de operações. Suponha-se que o operador tem que fechar duas válvulas, mas a ordem não é importante. Além disso, suponhamos que a primeira válvula leva 5 segundos para fechar, e a segunda válvula demora 1 segundo para fechar. Usando uma chamada assíncrona, o operador poderia emitir duas chamadas simultâneas, mesmo não havendo nenhuma garantia de que a segunda válvula (5s) fechará depois da primeira válvula (1s). Outro ponto importante neste modo de operação é o significativo ganho de performance quando não é necessária uma ordem de operações. O modo assíncrono é normalmente utilizado quando é necessário ler continuamente uma quantidade grande de dados que variam com o tempo, mas o cliente apenas é atualizado quando ocorre alteração significativa nos dados, como por exemplo, aplicações de armazenagem de dados históricos em sistemas supervisórios. Estas atualizações são correlacionadas com o *deadband* ou banda morta, definida como a porcentagem mínima de um determinado intervalo ou valor de medição que deve ser ultrapassado para que o dispositivo registre uma alteração. A Figura 1 mostra as duas formas de operação OPC, síncrona e assíncrona.

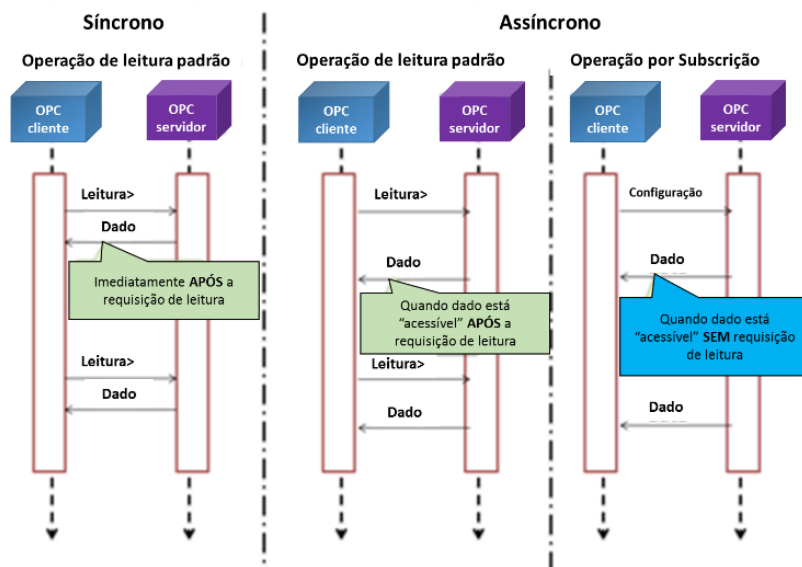


Figura 1 - Diagrama de leitura síncrono e assíncrono em OPC.

## 2.2 – MODBUS

O protocolo Modbus é uma estrutura de mensagens desenvolvida pela Modicon, hoje *Schneider Electric*, em 1979. O Modbus foi criado essencialmente para transferência de dados para configurar ou programar CLPs, sendo basicamente um protocolo responsivo, ou seja, a informação é transmitida através de consultas e respostas. É usada para estabelecer uma comunicação mestre-escravo entre dispositivos.

O mestre emite uma mensagem e, se o endereço de destino da mensagem corresponde ao endereço do dispositivo escravo, este responderá a mensagem. Todos os outros dispositivos escravos conectados ao barramento permanecem em silêncio até que recebam uma mensagem com seu endereço particular. O diagrama da Figura 2.2 demonstra esta implementação. O mestre deve aguardar o tempo suficiente para que o escravo receba, processe e responda a mensagem antes de enviar qualquer outra mensagem, pois se isso não acontecer, a sua próxima mensagem e a resposta anterior podem se sobrepor e ocasionar um erro de comunicação.

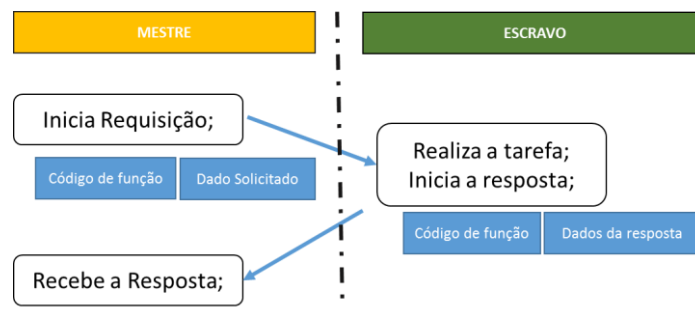


Figura 2.2 - Comunicação Mestre/Escravo.

Para possibilitar a integração no sistema de aquisição de dados foi utilizado uma ferramenta parte do software NI OPC Server da National Instruments que trabalha como uma ponte entre o protocolo OPC e o Modbus, mantendo assim os dados capturados através do Modbus disponíveis para qualquer cliente OPC ligado a rede.

### 3.0 - SINCRONISMO

Ao compartilhar dados entre máquinas em rede, manter os relógios das máquinas sincronizados é crucial, pois se cada máquina usar um horário diferente, torna-se impossível acompanhar as datas de modificações dos arquivos, ou até mesmo o *timestamp* de cada dado. Outro problema atrelado a falta de sincronismo é que o trabalho de ferramentas diversas de *backup* fica prejudicado, bem como os logs do sistema e outros recursos que dependem do horário. Para resolver o problema de sincronismo utilizam-se duas abordagens: sincronismo GPS local ou sincronismo via servidores de domínio público, ambos utilizando o protocolo NTP.

O protocolo NTP implementa um modelo de sincronização hierárquico distribuído. No topo encontram-se os servidores de tempo chamados *stratum 1*, computadores conectados diretamente a dispositivos conhecidos como "relógios de referência" (ou servidores *stratum 0*), de altíssima precisão. Tipicamente, estes dispositivos podem ser relógios atômicos, receptores GPS (*Global Positioning Systems*) ou receptores de rádio. Qualquer servidor NTP que tenha como referência de tempo um servidor *stratum 1* passa a ser um *stratum 2*, qualquer servidor NTP que tenha como referência de tempo um servidor *stratum 2* passa a ser um *stratum 3*, e assim por diante. O protocolo NTP leva em conta o *ping* entre as máquinas e outros fatores para fazer as atualizações de forma extremamente precisa. Diferenças de sincronismo entre os servidores são sempre da ordem de milésimos de segundo. Para o sistema implementado neste projeto foi utilizado o *software* NetTime, que atua como um cliente (SNTP) para ambientes Windows 95/98/Me/NT/2000/XP/Vista/7/8 e Server 2003/2008/2012.

### 4.0 - BANCO DE DADOS

Em essência, o sistema desenvolvido como produto desta etapa do projeto corresponde a um concentrador de dados, obtendo dados de diferentes sistemas de controle da planta, e concentrando-os em um histórico de dados. Conseqüentemente, os aspectos relativos ao armazenamento de tais dados exercem impacto direto neste sistema. Para fazer frente às necessidades neste quesito, empregou-se a tecnologia de bancos de dados.

Um sistema de banco de dados, usualmente denominado simplesmente banco de dados, é um sistema computadorizado especializado no armazenamento de informações e na disponibilização destas para manipulação pelo usuário, seja para inserir novas informações, removê-las, alterá-las ou simplesmente consultá-las. Tais informações podem ser de qualquer natureza de interesse para o usuário. No presente caso, correspondem a dados do processo de geração de energia da UEG Araucária (DATE, 2004). Sistemas de banco de dados oferecem vários benefícios aos usuários, dentre eles (DATE, 2004):

- Compartilhamento: As informações estão dispostas para todos os interessados que estejam autorizados a acessá-las;
- Redução da redundância: Os dados encontram-se centralizados, sem a necessidade de se manter cópias específicas para cada aplicação. Embora existam casos em que a redundância seja inevitável ou até desejada, os sistemas de banco de dados facilitam o seu controle;
- Redução de inconsistências: Quando os dados estão dispersos e redundantes, podem vir a surgir problemas de inconsistência, quando dois dados que deveriam ser equivalentes possuem valores distintos;
- Suporte a transações: Transações são conjuntos de operações em dados que devem ser tratados de maneira atômica, em que todas as operações são executadas ou nenhuma é. O exemplo clássico do uso de transações é a transferência monetária, em que ocorre o débito de uma quantia de uma conta e o crédito da mesma quantia em outra, correspondendo a duas operações de alteração do saldo das contas. Faz-se necessário que estas duas operações ocorram de forma conjugada, evitando o débito em uma conta sem o correspondente crédito ou o crédito sem o correspondente débito;
- Auxílio na garantia de integridade das informações: Sistemas de banco de dados oferecem recursos para tentar garantir a integridade dos dados, ou seja, que estes estejam corretos. Através de restrições de integridade inseridas no sistema, pode-se garantir que, por exemplo, não conste uma informação de que um funcionário trabalhou mais do que 24 horas em um único dia, ou que trabalhe em um departamento que não existe;
- Auxílio na segurança de informações: Sistemas de banco de dados permitem restringir o acesso a dados, ou a operações nos dados, de forma a evitar que dados confidenciais sejam acessados por pessoas não autorizadas.

### 5.0 - ARQUITETURA

A partir dos requisitos identificados em relação às necessidades da construção e implantação do sistema baseado em simulação e otimização a ser construído concebeu-se uma arquitetura geral composta dos seguintes elementos de *software* referentes à aquisição, armazenamento e uso de dados:

- Sistema de aquisição e armazenamento de dados: Programa de execução contínua que periodicamente adquire os dados do processo através do protocolo OPC e armazena as informações obtidas no banco de dados;

- Cliente de visualização de dados: Programa auxiliar que permite observar os dados do processo registrados no banco de dados, apresentando os valores atuais e traçando gráficos históricos de um conjunto de *tags* de interesse do usuário;
- Sistema de Banco de Dados: Sistema de banco de dados onde as informações adquiridas serão armazenadas;
- Interface de comunicação OPC: *Drivers* e aplicativos necessários para a realização da comunicação OPC com os sistemas de controle da usina.

A Figura 3 apresenta o diagrama UML (*Unified Modeling Language*) de componentes do sistema. Pode-se observar que o sistema de aquisição e armazenamento de dados depende do módulo de interface de comunicação OPC, de forma a possibilitar o acesso aos dados a serem adquiridos através deste protocolo. O sistema de aquisição depende também do sistema de banco de dados, pois é através dele que fará o armazenamento dos dados adquiridos. Por sua vez, o cliente de visualização de dados também possui uma relação de dependência com o sistema de banco de dados pois obtém dele os dados a serem apresentados ao usuário.

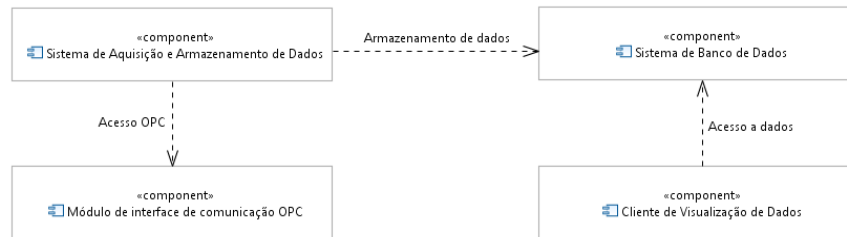


Figura 3 - Diagrama UML de componentes do Sistema de aquisição e armazenamento de dados.

No caso do componente “Sistema de Aquisição e Armazenamento de Dados”, componente principal do sistema, este está organizado tal como representado no diagrama UML de pacotes da Figura 4. O módulo central do sistema, denominado “Núcleo” realiza o controle de execução do sistema como um todo, e interage com todos os demais módulos. O módulo de banco de dados é responsável por encapsular a interação com o sistema de banco de dados e efetuar neste o armazenamento das aquisições. O módulo de comunicação com a usina é responsável por estabelecer as conexões com os diferentes sistemas de controle desta, e realizar as aquisições dos dados de acordo com as solicitações do núcleo. O módulo de log é responsável por registrar as entradas informadas pelo módulo do núcleo no arquivo de log, e gerenciar o tamanho deste arquivo, removendo entradas antigas caso necessário. O módulo de configuração é responsável por ler as configurações do sistema dos arquivos apropriados e disponibilizá-las aos demais módulos. Finalmente, o módulo de acesso direto é responsável por prover um mecanismo de acesso às informações do estado recente da planta à sistemas externos, em especial o sistema de eficiência a ser desenvolvido no decorrer do projeto, sem a necessidade que estes realizem consultas ao banco de dados.

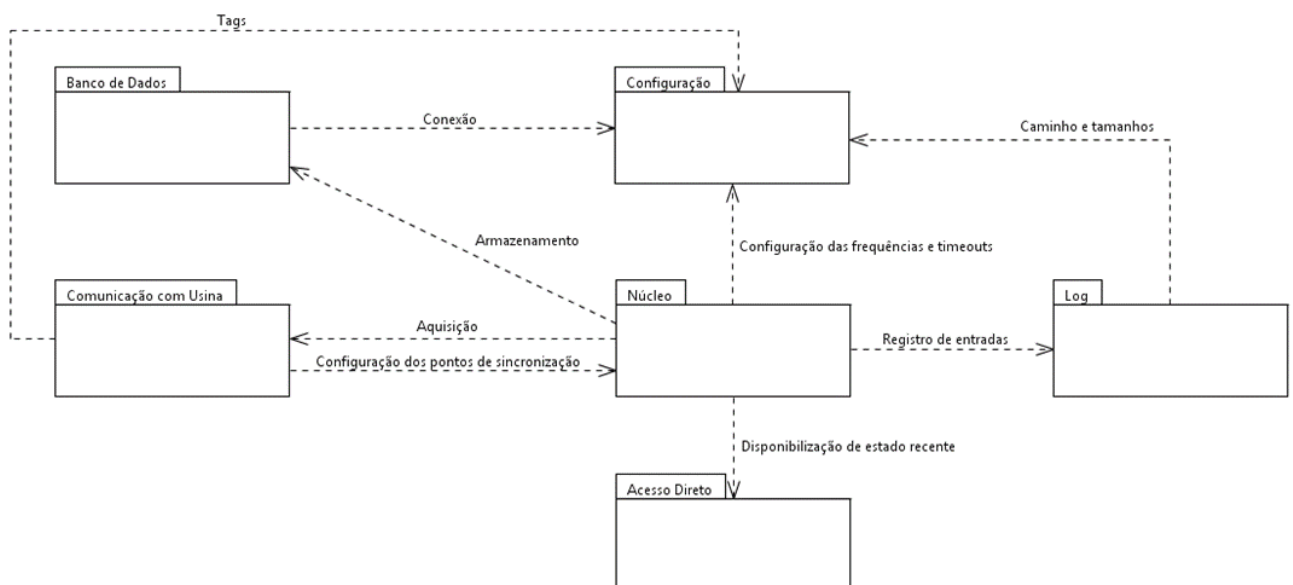


Figura 4 - Diagrama UML de pacotes do componente de aquisição e armazenamento de dados.

Todos estes componentes são implantados em uma única máquina servidor, o equipamento Dell PowerEdge R530 com uma licença Windows Server 2012, implantado em um *rack* na sala de servidores da UEG Araucária. Além da arquitetura de componentes de *software* do sistema, é necessário explicitar a arquitetura de rede e comunicação com os sistemas de controle da usina, dos quais são obtidos os dados a serem armazenados. As conexões básicas entre os subsistemas, que são sistemas autônomos de medição, foram realizadas por meio dos protocolos Modbus RTU, Modbus TCP e OPC. Os subsistemas que interagem com o sistema de aquisição, bem como a forma de tal interação e sua conexão lógica com o servidor citado (nomeado Servidor Lactec), são apresentados na Figura 5.

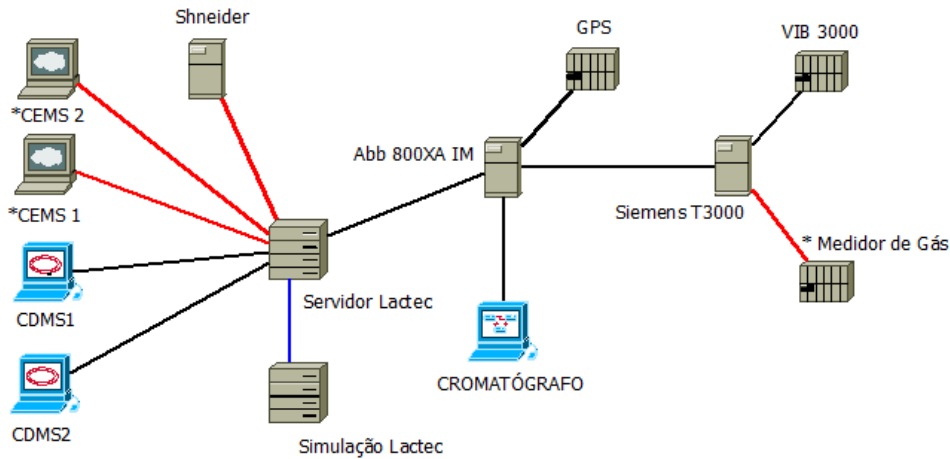


Figura 5 - Diagrama de conexão lógica dos subsistemas.

A Figura 6 apresenta o diagrama UML de atividades simplificado do sistema, abordando sua arquitetura dinâmica, isto é, seu comportamento. Ao ser iniciada sua execução, o sistema passará a desempenhar a atividade de inicialização. Nesta etapa, todas as informações necessárias para a execução das demais atividades são obtidas, e a configuração do sistema é realizada. Na eventualidade de algum problema ocorrer durante esta atividade, o sistema o registrará no arquivo de log e será finalizado. Caso contrário, o sistema iniciará quatro processos simultâneos, cada qual responsável pela execução de uma atividade, a saber, a aquisição de dados de processo da planta, o registro histórico das aquisições, a disponibilização do *snapshot* do estado recente da planta e o registro de eventos e erros no arquivo de log.

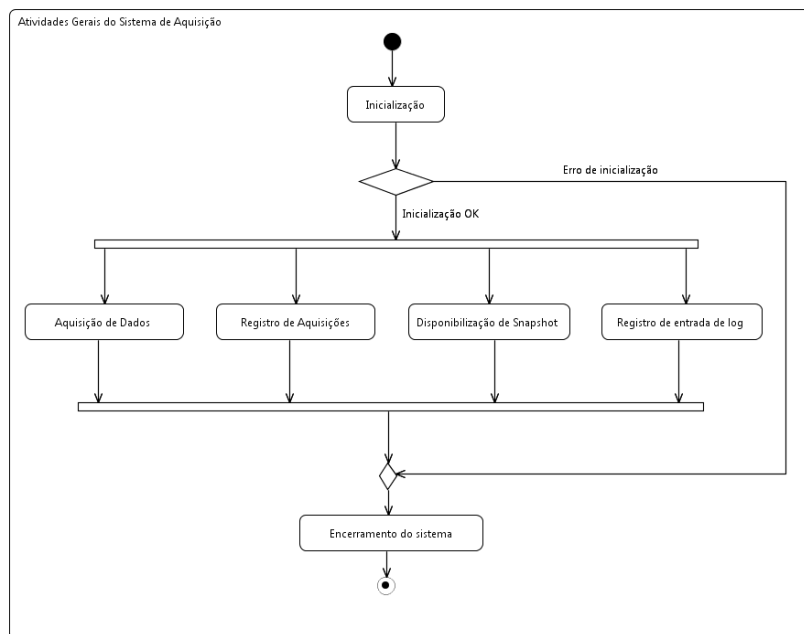


Figura 6 - Diagrama UML de atividades simplificado do sistema de aquisição.

## 6.0 - RESULTADOS

Nesta seção são apresentados os resultados do sistema de aquisição bem como o resultado da identificação de modelos utilizando os dados reais de uma turbina a gás *Heavy Duty*.

O sistema de aquisição construído vem sendo utilizado desde o final de 2015, obtendo mais de 500 gigabytes de dados. De particular relevância para o projeto são os dados das partidas e paradas dos grupos geradores realizados durante seus *tunnings*, e as rampas de tomada e retirada de carga correspondentes, empregados para criação, ajustes e testes dos modelos matemáticos criados.

Para a validação do processo de construção de modelos matemáticos a partir de Inteligência Computacional a partir de dados reais foi empregada uma rede neural nebulosa Fuzzy clusterizada usando Fuzzy c-means e com pesos treinados por meio da pseudo-inversa de Moore-Penrose. A rede Neuro Fuzzy foi configurada com cinco neurônios, sendo que foram executadas 100 iterações e 50 iterações com critério de parada  $1e-3$  para a rede Fuzzy (BEZDEK, JAMES, 1981). O treinamento e validação das técnicas foi realizado utilizando 90% dos dados para treinamento e 10 % para validação (A partir da amostra 13500).

A Figura 7 apresenta o intervalo de dados da potência ativa de uma unidade térmica visando avaliar o comportamento do modelo preditivo de  $NO_x$  quando a turbina a gás opera em “plena carga”, ou seja, próximo dos 160 MW.

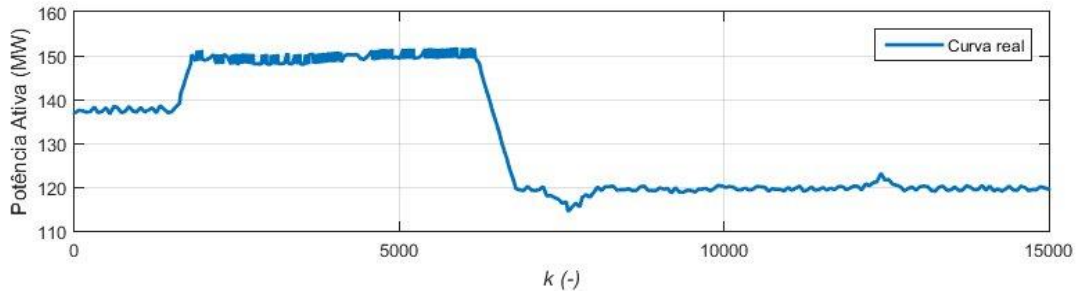


Figura 7 - Gráfico de potência ativa proveniente de uma turbina a gás.

O resultado do modelo matemático que representado o comportamento das emissões de  $NO_x$  de uma turbina *Heavy Duty* a gás natural operando com potência ativa no intervalo de 120 a 150 MW. Tal modelo foi construído utilizando 20 variáveis cujos dados foram obtidos a partir do sistema de aquisição descrito anteriormente. A Figura 8 apresenta as curvas de  $NO_x$  corrigido, cujo dados são provenientes do sistema de aquisição, e a curva obtida a partir do modelo matemático obtido a partir de simulação livre.

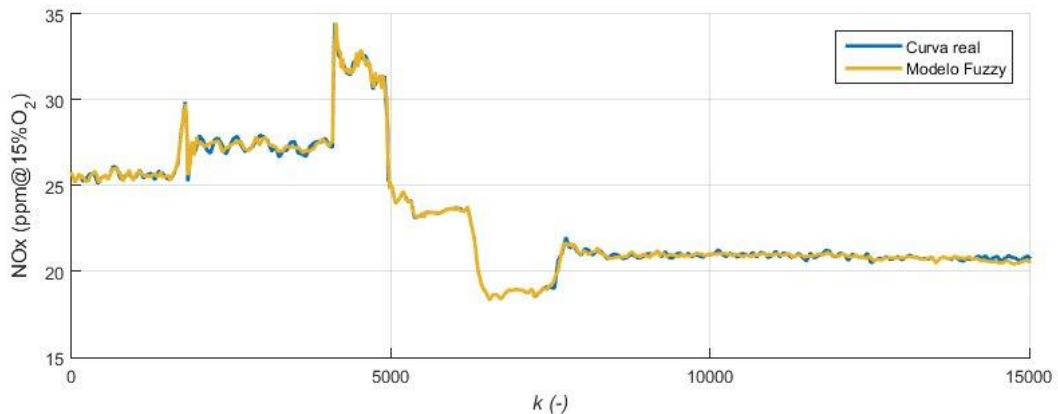


Figura 8 – Comportamento  $NO_x$  adquirido (dado real) e previsto.

Para a avaliação dos resultados foram empregados o Erro Médio Quadrado (MSE – *Mean Squared Error*) e o Erro Percentual Absoluto Médio (MAPE – *Mean Absolute Percentual Error*), bem como o tempo de treinamento das técnicas. Vale observar que a máquina usada para as simulações estava sendo usada para outras tarefas durante o processo e as configurações da mesma são processador Intel Core i7 2.93GHz, 10GB de RAM, 1333MHz e sistema operacional 64 bits. A Tabela 1 apresenta os erros de treinamento e validação do modelo identificado bem como seu tempo de treinamento.

Tabela 1 – Resultados de  $NO_x$ .

Treinamento		Validação		Tempo de treinamento (s)
MSE	MAPE	MSE	MAPE	
0,023957	0,423909	0,03422	0,740898	45,271

## 7.0 - CONCLUSÃO

O sistema de aquisição e armazenamento de dados da planta foi desenvolvido e implantado com sucesso.

O desenvolvimento do sistema seguiu as etapas clássicas do desenvolvimento de *software*, começando com o levantamento dos requisitos que definiam o escopo funcional e não-funcional do sistema. Estes foram verificados pela equipe e após aprovados, orientaram o projeto do sistema de aquisição e sua posterior implementação. No decorrer do projeto do sistema, algumas adaptações aos requisitos foram impostas pelo contexto de implantação do sistema, em especial as questões de segurança da informação da usina, adaptações estas que foram discutidas e acordadas entre a equipe. A implementação, por sua vez, foi realizada de acordo com o projeto, e este refinado no decorrer da implementação, em um processo iterativo buscando a qualidade do sistema. Tal processo, entretanto, buscou manter a rastreabilidade entre os requisitos, projeto e implantação, de forma a mantê-los consistentes.

Uma das principais preocupações no decorrer do desenvolvimento do sistema foi em relação à performance, tanto no que tange o desempenho do banco de dados frente ao montante e frequência de informações a serem registradas quanto às questões relacionadas a redes e comunicação OPC. Sendo assim, testes foram realizados durante o desenvolvimento para avaliar o impacto das decisões de arquitetura e de tecnologias a serem empregadas. Após o desenvolvimento, mais testes foram realizados com o intuito de comprovar o funcionamento do sistema e que desempenhava a contento.

## 8.0 - REFERÊNCIAS BIBLIOGRÁFICAS

- (1) BEZDEK, JAMES C. (1981). *Pattern Recognition with Fuzzy Objective Function Algorithms*.
- (2) DAMAS, L. SQL - Structured Query Language. Tradução de Elvira Maria Antunes. 6ª. ed. Rio de Janeiro: LTC, 2007.
- (3) DATE, C. J. Introdução a sistemas de bancos de dados. Tradução de Daniel Vieira. 8. ed. Rio de Janeiro: Elsevier Editora Ltda., 2004.
- (4) DEON REYNDERS, S. M. E. W. Practical industrial data communications: best practice techniques (practical professional) 1st Edition. [S.l.]: Butterworth-Heinemann, 2005.
- (5) KLINE, K. E.; KLINE, D.; HUNT, B. SQL - O guia essencial. Tradução de Betina de Oliveira Lima e Paloma Pontes. Rio de Janeiro: Alta Books, 2010.
- (6) MILANI, A. PostgreSQL - Guia do programador. São Paulo: Novatec Editora Ltda., 2008.
- (7) WOLFGANG MAHNKE, S.-H. L. M. D. OPC unified architecture. Verlag Berlin Heidelberg: Springer, 2009.

## 9.0 - DADOS BIBLIOGRÁFICOS

	<p><b>Autor de Contato:</b> Luís Gustavo  <b>Local e ano de nascimento:</b> Ponta Grossa – PR/ 1989  <b>Local e ano de graduação / pós-graduação:</b> Curitiba-PR/2004  <b>Experiência profissional:</b> Engenheiro Eletricista – Ênfase Eletrônica e Telecomunicações pela UFPR. Pesquisador nos Institutos Lactec.</p>
<p><b>Autor 2:</b> Eduardo Massashi Yamao  <b>Local e ano de nascimento:</b> Curitiba, 1985  <b>Local e ano de graduação / pós-graduação:</b> Curitiba, 2009/2016  <b>Experiência profissional:</b> Mestrado em Engenharia Elétrica pela UFPR. Experiência de 6 anos como Pesquisador nos Institutos Lactec atuando nas áreas de simulação, treinamento, Usinas Hidrelétricas, Termelétricas e Eólicas.</p>	<p><b>Autor 3:</b> Fabio Yukio Nara  <b>Local e ano de nascimento:</b> Santo Antônio da Platina, 1984  <b>Local e ano de graduação / pós-graduação:</b> Curitiba, 2009  <b>Experiência profissional:</b> Graduado em Engenharia Mecânica pela UFPR. Atua como pesquisador nos Institutos Lactec na área de projetos mecânicos, modelagem e simulação de sistemas mecânicos e simulação aplicada a mecânica dos fluidos por fluidodinâmica computacional (CFD).</p>



<p><b>Autor 4:</b> Felipe Taborda Ribas Tovar  <b>Local e ano de nascimento:</b> Curitiba-PR, 1980  <b>Local e ano de graduação / pós-graduação:</b> Curitiba,2014  <b>Experiência profissional:</b> Graduado em Engenharia de Computação pela PUC-PR. Atua na área de engenharia e desenvolvimento de software com experiência em desenvolvimento de sistemas de treinamento baseados em simulação.</p>	<p><b>Autor 5:</b> João Borda  <b>Local e ano de nascimento:</b> São Paulo-SP,1959  <b>Local e ano de graduação / pós-graduação:</b> Curitiba,1984  <b>Experiência profissional:</b> Graduado em Engenharia Mecânica pela UFPR. Atua como gestor de projetos com experiência nas áreas de processos de usinagem, estamparia, solda robotizada, automobilística, papel &amp; celulose e construção civil.</p>
<p><b>Autor 6:</b> Flávio Chiesa  <b>Local e ano de nascimento:</b> Curitiba-PR, 1959  <b>Local e ano de graduação / pós-graduação:</b> Curitiba,1981/ Florianópolis,1994/ Curitiba,2002  <b>Experiência profissional:</b> Graduado em Engenharia Mecânica (UFPR), também possui pós-graduação em Engenharia de Produção pela UFSC e Mestrado em Engenharia Mecânica pelaPUC-PR. Já atuou em diversas áreas relacionadas a geração de energia e engenharia térmica. Desde 2007 é diretor técnico da Usina Termoelétrica de Araucária (UEGA) de 480MW a ciclo combinado.</p>	<p><b>Autor 7:</b> Victor Manuel Lopes Santos  <b>Local e ano de nascimento:</b> Cabo Verde, 1960  <b>Local e ano de graduação / pós-graduação:</b> Curitiba,1992  <b>Experiência profissional:</b> Graduado em Engenharia Elétrica, trabalha na coordenação e gerencia técnica da usina termoelétrica de Araucária (UEGA).</p>
<p><b>Autor 8:</b> André da Silva Orlandi  <b>Local e ano de nascimento:</b> Cambé-PR, 1986  <b>Local e ano de graduação / pós-graduação::</b> Curitiba, 2008/ Curitiba,2009  <b>Experiência profissional:</b> Graduado em Engenharia Elétrica pela UFSC. Atualmente é Engenheiro de Manutenção de Equipamentos Eletrônicos - Copel Geração e Transmissão S.A. em usinas de geração de energia.</p>	