

**NANOTRACK - SISTEMA INTELIGENTE DE  
GERENCIAMENTO DE DADOS DE SÍNTESE DE  
NANOESTRUTURAS**



ANDRÉ SOUSA FIGUEIREDO SILVA

**NANOTRACK - SISTEMA INTELIGENTE DE  
GERENCIAMENTO DE DADOS DE SÍNTESE DE  
NANOESTRUTURAS**

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação do Instituto de Ciências Exatas da Universidade Federal de Minas Gerais. Departamento de Ciência da Computação como requisito parcial para a obtenção do grau de Mestre em Ciência da Computação.

ORIENTADOR: OMAR PARANAÍBA VILELA NETO

Belo Horizonte

Março de 2013

© 2013, André Sousa Figueiredo Silva.  
Todos os direitos reservados.

Silva, André Sousa Figueiredo  
S586n NanoTrack - Sistema Inteligente de Gerenciamento  
de Dados de Síntese de Nanoestruturas / André Sousa  
Figueiredo Silva. — Belo Horizonte, 2013  
xxii, 78 f. : il. ; 29cm

Dissertação (mestrado) — Universidade Federal de  
Minas Gerais. Departamento de Ciência da  
Computação

Orientador: Omar Paranaíba Vilela Neto

1. Computação - Teses. 2. Nanotecnologia - Teses.  
3. Inteligência Computacional - Teses. I. Orientador  
II. Título.

CDU 519.6\*61(043)





UNIVERSIDADE FEDERAL DE MINAS GERAIS  
INSTITUTO DE CIÊNCIAS EXATAS  
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

## FOLHA DE APROVAÇÃO

NanoTrack - Sistema inteligente de gerenciamento de dados de síntese de nanoestruturas

**ANDRÉ SOUSA FIGUEIREDO SILVA**

Dissertação defendida e aprovada pela banca examinadora constituída pelos Senhores:

A blue ink signature of Prof. Omar Paranaíba Vilela Neto, written in a cursive style.

PROF. OMAR PARANAÍBA VILELA NETO - Orientador  
Departamento de Ciência da Computação - UFMG

A blue ink signature of Dra. Alessandra Conceição Faria Aguiar Campos, written in a cursive style.

DRA. ALESSANDRA CONCEIÇÃO FARIA AGUIAR CAMPOS  
Bolsista Pós-Doc/ ICB - UFMG

A blue ink signature of Prof. Paulo Sérgio Soares Guimarães, written in a cursive style.

PROF. PAULO SÉRGIO SOARES GUIMARÃES  
Departamento de Física - UFMG

A blue ink signature of Prof. Sérgio Vale Aguiar Campos, written in a cursive style.

PROF. SÉRGIO VALE AGUIAR CAMPOS  
Departamento de Ciência da Computação - UFMG

Belo Horizonte, 22 de março de 2013.



# Agradecimentos

Agradeço a Deus pela vida. Aos meus pais, Marden e Neide, por minha formação e apoio incondicional. Aos meus irmãos, Thiago e Diogo, pela amizade e incentivos. A Maíra, pelo carinho, amor e companhia. Aos familiares e amigos, por torcer pelo meu sucesso. Ao meu orientador, Omar Paranaíba, um grande professor, cientista e amigo que me ajudou durante todo o período com otimismo e confiança. Ao laboratório LUAR, especialmente ao Sérgio, Alessandra e Paulo, por confiar em nosso projeto e sempre ajudar com muita disposição. Aos integrantes do laboratório LECOM, sempre apoiando e incentivando na convivência diária, especialmente ao Artur Melo, que contribuiu no desenvolvimento deste trabalho. Ao LabSem (PUC-Rio) e Laboratório de Nanomateriais que forneceu dados fundamentais para os estudos de caso. Aos funcionários da secretaria do Departamento de Ciência da Computação pelo apoio com a parte burocrática. A CAPES pela bolsa. A todos que, direta ou indiretamente, contribuíram para a realização deste trabalho.



*“Vencer, vencer, vencer. Este é o nosso ideal!”*  
(Letra do hino do Clube Atlético Mineiro)



# Resumo

O interesse em Nanociência e Nanotecnologia vem aumentando devido à possibilidade de desenvolvimento de novos materiais com propriedades aprimoradas e inovadoras. Segundo os cientistas, a Nanotecnologia será capaz de mudar a natureza de quase tudo que já foi feito pelos seres humanos, gerando um impacto em áreas como medicina, engenharias, telecomunicações, energia, computação, etc. Entretanto, a Nanotecnologia carece de ferramentas computacionais que apoiem o pesquisador em suas atividades de pesquisa. Normalmente, as tarefas envolvidas no desenvolvimento de nanoestruturas não são simples de serem executadas. Vários fatores tecnológicos e econômicos contribuem para esta complexidade. Para a fabricação de qualquer composto é essencial o conhecimento sobre o material que está sendo construído e em compostos nanométricos não é diferente. Porém, a obtenção destes dados é dificultada devido à precariedade com que as informações dos experimentos são armazenadas ou à restrição de acesso aos dados de outros grupos. Com isto, a descoberta de conhecimento pelo pesquisador é restringida dificultando o desenvolvimento de nanomateriais inovadores. Os LIMS (Laboratory Information Management Systems) são sistemas desenvolvidos com o objetivo de facilitar o gerenciamento e armazenamento de dados de laboratórios e já vem sendo bastante utilizado por outras áreas. Baseado na plataforma Flux<sup>®</sup>, neste trabalho foi desenvolvido a ferramenta NanoTrack, um LIMS para apoio à Nanotecnologia. Além do gerenciamento, a ferramenta provê um ambiente de simulação capaz de apoiar os pesquisadores na descoberta de conhecimento. Este ambiente é formado por plugins que são incorporados à ferramenta e são compostos por algoritmos de inferência e otimização baseados em Redes Neurais Artificiais e Algoritmos Genéticos, respectivamente. Neste trabalho foram desenvolvidos três estudos de casos de laboratórios de Nanotecnologia: Pontos Quânticos, Laboratório de Semicondutores e Nanotubos de Carbono. Os resultados obtidos demonstram o poder da ferramenta no gerenciamento dos dados e a capacidade dos algoritmos computacionais na descoberta de parâmetros importantes para a síntese de nanoestruturas e nanodispositivos.

**Palavras-chave:** Nanotecnologia, Nanociência, Nanoinformática, LIMS, Workflow, Rede Neural Artificial, Programação Genética, Ambiente de simulação.



# Abstract

The interest in Nanoscience and Nanotechnology has increased due to the possibility of developing new materials with improved and innovative properties. According to the scientists, Nanotechnology will be able to change the nature of almost everything that has been done by humans, generating an impact in areas such as medicine, engineering, telecommunications, energy, computing, etc. However, Nanotechnology lacks computational tools that support the researcher in his research activities. Usually, the tasks involved in the development of nanostructures are not simple to be executed. Several technological and economic factors contribute to this complexity. For the manufacture of any compound, the knowledge about the material that is being built is essential. For nanosized compounds there is no difference and one should have knowledge about the nanosized compound. However, obtaining the required data is difficult due to the precariousness with which information of the experiments are stored and also due to restriction about the data from other groups. Because of this, the knowledge discovery by the researcher is restricted, and this is something that hampers the development of innovative nanomaterials. The LIMS (Laboratory Information Management Systems) are systems developed to facilitate the storage and management of data from laboratories and are already being widely used in other areas. Based on the Flux® platform, we have developed the NanoTrack tool, that is a LIMS for Nanotechnology support. In addition to the management, the tool provides a simulation environment that supports researchers with knowledge discovery. This environment consists of plugins that are incorporated into the tool. These plugins are composed by inference and optimization algorithms, based on Artificial Neural Networks and Genetic Algorithms, respectively. This work presents three developed case studies of the Nanotechnology laboratories: Quantum Dots, Semiconductor Laboratory and Carbon Nanotubes. The results obtained demonstrate the tool power in data management and its computer algorithms capacity for discovering important parameters for the synthesis of nanostructures and nanodevices.

**Keywords:** Nanotechnology, Nanoscience, Nanoinformatics, LIMS, Workflow, Artificial Neural Networks, Genetic Programming, Simulation Environment.

# Lista de Figuras

2.1	Relação equivalente de escalas. . . . .	7
2.2	Escalas abaixo de 1 metro. . . . .	8
2.3	Exemplo de uma rede neural simples com duas camadas intermediárias. . .	14
2.4	X-Informática. . . . .	17
2.5	Paradigmas da ciência. . . . .	18
4.1	Interface de login da ferramenta NanoTrack. . . . .	27
4.2	Esboço da modelagem do Laboratório de Semicondutores (PUC-Rio). . . .	29
4.3	Processo de modelagem de um laboratório para a ferramenta NanoTrack. .	30
4.4	Estrutura genérica de interação entre o NanoTrack, plugins e banco de dados.	31
4.5	Atividade de consulta externa no NanoTrack. . . . .	34
4.6	Resultado da saída do plugin de consulta externa. . . . .	34
4.7	Protótipo da interação entre NanoTrack e Plugin de treinamento da rede neural. . . . .	36
4.8	Protótipo da interação entre NanoTrack e Plugin para utilização de uma rede neural já treinada. . . . .	37
5.1	Workflow de experimentação de Pontos Quânticos. . . . .	40
5.2	Interface do Together Workflow Editor com o workflow de Pontos Quânticos sendo codificado para XPDL. . . . .	41
5.3	Trinta atributos que compõem a atividade de síntese do workflow de Pontos Quânticos. . . . .	41
5.4	Interface do Nanotrack exibindo os dados cadastrados em uma atividade de síntese. . . . .	42
5.5	Utilização da rede neural treinada dentro do ambiente de simulação do Nanotrack. . . . .	46
5.6	Resultado da rede neural nas etapas de treinamento, validação e teste para a saída de Densidade. . . . .	48

5.7	Resultado da rede neural nas etapas de treinamento, validação e teste para a saída de Altura Média. . . . .	49
5.8	Resultado da rede neural nas etapas de treinamento, validação e teste para a saída de Desvio Padrão. . . . .	50
5.9	Workflow do Laboratório de Semicondutores (PUC-Rio). . . . .	51
5.10	Interface do Together Workflow Editor durante a codificação da atividade de Raio-X no workflow do LabSem. . . . .	53
5.11	Oitenta atributos que compõem a atividade de síntese do workflow de LabSem. . . . .	53
5.12	Interface do Nanotrack durante a atividade de síntese de camada do workflow do LabSem. . . . .	54
5.13	Resultado da rede neural nas etapas de treinamento, validação e teste para a concentração de Gálio. . . . .	58
5.14	Resultado da rede neural nas etapas de treinamento, validação e teste para a concentração de Alumínio. . . . .	59
5.15	Indivíduo evoluiu durante a execução do algoritmo genético. . . . .	61
5.16	Interface do “Simulador Quaternário” no NanoTrack durante o preenchimento dos parâmetros de entrada do simulador. . . . .	62
5.17	Estrutura do plugin mostrando as interações entre o NanoTrack, plugin e o Matlab. . . . .	63
5.18	Ambiente de simulação do workflow representado pela atividade “Simulador Quaternário” . . . . .	64
5.19	Workflow do Laboratório de Nanotubos de Carbono. . . . .	65
5.20	Interface do Fluxar com o workflow de Nanotubos de Carbono sendo codificado para XPDL. . . . .	67
5.21	Atributos da atividade de síntese do workflow de Nanotubos de Carbono. . . . .	68
5.22	Oitenta atributos que compõem a atividade de síntese do workflow de LabSem. . . . .	69
6.1	Modelo de um sistema integrado de inferência. . . . .	73

# Lista de Tabelas

4.1	Exemplo de base de dados . . . . .	33
4.2	Resultado obtido com utilização da consulta desenvolvida . . . . .	33
5.1	Lista de atividades identificadas durante a criação do workflow de Pontos Quânticos. . . . .	40
5.2	Tabelas externas criadas para as atividades do workflow de Pontos Quânticos	42
5.3	Lista de atributos de entrada e saída utilizados na rede neural. . . . .	44
5.4	Quantidade de neurônios em cada camada da Rede Neural . . . . .	46
5.5	Lista de atividades identificadas durante a criação do workflow do Laboratório de Semicondutores. . . . .	52
5.6	Tabelas externas criadas para as atividades do workflow do LabSem . . . . .	54
5.7	Lista de atributos de entrada e saída utilizados na rede neural. . . . .	56
5.8	Composições de quaternários carentes de documentação. . . . .	60
5.9	Bons resultados obtidos com algoritmo genético. . . . .	61
5.10	Lista de atividades identificadas durante a criação do workflow de Nanotubos de Carbono. . . . .	65
5.11	Tabelas externas criadas para as atividades do workflow de Nanotubos de Carbono . . . . .	67



# Lista de Siglas

**ABNT** Associação Brasileira de Normas Técnicas

**BPL** Boas Práticas de Laboratório

**BPMN** Business Process Modeling Notation

**HTML** HyperText Markup Language

**INMETRO** Instituto Nacional de Metrologia, Qualidade e Tecnologia

**ISO** International Organization for Standardization

**LabSem** Laboratório de Semicondutores

**LIMS** Laboratory Information Management Systems

**LUAR** Laboratório de Universalização do Acesso

**MBAR** Milibar

**NM** Nanômetro

**PGLIQ** Programação Genética Linear com Inspiração Quântica

**QD/cm<sup>2</sup>** Quantum Dots por centímetro quadrado

**SCCM** Standard Cubic Centimeters per Minute

**SCUFL** Simple Conceptual Unified Flow Language

**SQL** Structured Query Language

**TWE** Together Workflow Editor

**XML** eXtensible Markup Language

**XPDL** XML Process Definition Language

**WfMC Workflow Management Coalition**



# Sumário

<b>Agradecimentos</b>	<b>vii</b>
<b>Resumo</b>	<b>xi</b>
<b>Abstract</b>	<b>xiii</b>
<b>Lista de Figuras</b>	<b>xv</b>
<b>Lista de Tabelas</b>	<b>xvii</b>
<b>Lista de Siglas</b>	<b>xix</b>
<b>1 Introdução</b>	<b>1</b>
1.1 Motivação . . . . .	2
1.2 Objetivo . . . . .	3
1.3 Resumo dos resultados . . . . .	4
1.4 Contribuições . . . . .	5
1.5 Organização da dissertação . . . . .	6
<b>2 Fundamentos</b>	<b>7</b>
2.1 A escala nano . . . . .	7
2.2 Sistemas de Gerenciamento de Informação de Laboratórios - LIMS . . . . .	8
2.2.1 Gerenciamento da explosão da informação . . . . .	10
2.2.2 Garantia da qualidade . . . . .	10
2.2.3 Maior eficiência e eficácia na execução de atividades do laboratório	11
2.3 Workflow . . . . .	12
2.3.1 XPDL - XML Process Definition Language . . . . .	12
2.4 Redes Neurais Artificiais . . . . .	13
2.4.1 Tratamento de dados . . . . .	15
2.4.2 Seleção de amostras . . . . .	15

2.4.3	Treinamento . . . . .	15
2.5	Nanoinformática . . . . .	16
<b>3</b>	<b>Trabalhos relacionados</b>	<b>21</b>
3.1	LIMS e Workflows . . . . .	21
3.2	Inferência de nanoestruturas . . . . .	23
<b>4</b>	<b>LIMS em nanotecnologia</b>	<b>27</b>
4.1	Modelagem . . . . .	28
4.2	Migração de dados . . . . .	30
4.3	Plugins . . . . .	31
4.3.1	Interação: Plugin x NanoTrack . . . . .	32
4.3.2	Consulta Externa . . . . .	32
4.3.3	Plugin de Inferência (Rede Neural) . . . . .	35
<b>5</b>	<b>Estudos de casos</b>	<b>39</b>
5.1	Pontos Quânticos . . . . .	39
5.1.1	Modelagem . . . . .	39
5.1.2	Importação de dados . . . . .	42
5.1.3	Consulta Externa . . . . .	43
5.1.4	Ambiente de simulação . . . . .	43
5.2	Laboratório de Semicondutores (PUC-Rio) . . . . .	50
5.2.1	Modelagem . . . . .	51
5.2.2	Importação de dados . . . . .	54
5.2.3	Ambiente de simulação . . . . .	55
5.3	Nanotubos de Carbono (Física-UFMG) . . . . .	63
5.3.1	Modelagem . . . . .	64
<b>6</b>	<b>Conclusão e trabalhos futuros</b>	<b>71</b>
6.1	Trabalhos futuros . . . . .	72
	<b>Referências Bibliográficas</b>	<b>75</b>

# Capítulo 1

## Introdução

O avanço tecnológico observado nas últimas décadas tem propiciado a geração, o armazenamento e a divulgação de uma grande quantidade de dados. Diversas áreas de pesquisa e desenvolvimento têm buscado alternativas eficientes para lidar com esta grande onda de informações, de forma a filtrar e trabalhar com os dados que são realmente essenciais.

Recentemente, o interesse em nanociência e nanotecnologia vem aumentando devido ao grande potencial que pode propiciar benefícios em diversas áreas e a possibilidade de desenvolvimento de novos materiais com propriedades aprimoradas e inovadoras. Segundo os cientistas, a nanotecnologia será capaz de mudar a natureza de quase tudo que já foi feito pelos seres humanos, gerando um impacto em áreas como medicina, engenharia, telecomunicações, energia, informática, dentre outros [Dias et al., 2011]. Como consequência, poderemos ter o desenvolvimento de novos medicamentos, despoluição da água e do ar, aperfeiçoamento das tecnologias de comunicação e informação, produção eficiente de energia, desenvolvimento de materiais mais leves e resistentes, etc.

A nanociência e a nanotecnologia podem ser definidas como o estudo, projeto, caracterização, produção e aplicação de estruturas, dispositivos e sistemas controlando a forma e o tamanho na escala atômica e molecular (normalmente definida entre 1 e 100 nm, onde 1 nm é igual a  $10^{-9}$  metros), tirando proveito das propriedades diferentes daquelas encontradas em materiais de maior escala. A nanociência e a nanotecnologia são áreas genuinamente interdisciplinares e têm possibilitado a colaboração entre pesquisadores de áreas distintas, permitindo a troca de informações, ferramentas e técnicas. Não há dúvidas que a nanotecnologia revolucionará quase tudo aquilo realizado pelo ser humano, até mesmo os computadores.

Esta nova revolução tecnológica já conta com o apoio fundamental obtido a partir

do avanço da computação. Hoje, todas as áreas do conhecimento fazem o uso da computação para automatizar e facilitar a pesquisa e desenvolvimento. Isto não é diferente com a nanotecnologia. O desenvolvimento de modelos computacionais de sistemas químicos, físicos e biológicos, além do avanço na capacidade de processamento dos computadores, tem permitido aos pesquisadores a modelagem e simulação de possíveis nanomateriais, nanodispositivos e aplicações. Este apoio dos sistemas de computação à nanociência e nanotecnologia recebe o nome de *Nanotecnologia Computacional*. Além das simulações e modelagem de nanoestruturas, o apoio computacional à nanociência e à nanotecnologia também pode ser dado por uma infraestrutura de armazenamento, interpretação, manipulação, compartilhamento e divulgação de dados e informações de pesquisa. Esta é uma nova área de pesquisa dentro da nanotecnologia computacional e tem sido chamada de *Nanoinformática*.

## 1.1 Motivação

A nanotecnologia carece de ferramentas computacionais que apóiam o pesquisador em suas atividades de pesquisa [de la Iglesia et al., 2011]. Normalmente, as tarefas envolvidas na síntese de nanoestruturas não são simples de serem executadas. A obtenção de dados para a fabricação de qualquer composto nanométrico é dificultada por fatores tecnológicos e econômicos e esta falta de dados precisos sobre os experimentos com nanomateriais restringe a descoberta de conhecimento e dificulta a criação de nanomateriais inovadores [de la Iglesia et al., 2011]. Alguns fatores que prejudicam o avanço nas pesquisas são: falta de estrutura apropriada e dedicada aos experimentos, o tempo despendido na experimentação é grande, informações associadas ao experimento armazenadas de forma precária ou mesmo inexistente, dentre outros.

Lidar com elementos na escala nanométrica não é trivial e exige um investimento alto. Equipamentos com esta precisão não são comuns em quaisquer laboratórios de pesquisa e, por isto, pesquisadores podem ter dificuldade no acesso a tais equipamentos. Além disto, existem outros custos, tais como a preparação das amostras e tempo de esforço do pesquisador. O prévio conhecimento da estrutura pelo pesquisador é um fator muito importante. Analisar uma estrutura por completo pode ser impraticável e, quando o pesquisador já possui um conhecimento da estrutura, ele consegue direcionar o seu foco para regiões que ele entende ser o mais provável de encontrar o que procura.

Para a fabricação de qualquer composto é essencial o conhecimento sobre o material que está sendo construído e em compostos nanométricos não é diferente. Porém, a obtenção destes dados é dificultada pelos fatores citados acima. Esta falta de dados so-

bre os nanomateriais, devido ao armazenamento precário de experimentos anteriores ou restrição de acesso aos dados de outros grupos, impede a descoberta de conhecimento pelo pesquisador, dificultando a criação de nanomateriais inovadores. Neste contexto, as ferramentas computacionais podem exercer um importante papel no auxílio às novas descobertas.

## 1.2 Objetivo

O objetivo principal deste trabalho é construir um sistema para gerenciamento de dados de experimentos em nanotecnologia, com uma plataforma de apoio ao pesquisador que possua funcionalidades adequadas à captura e análise de dados desta natureza, como atividades de síntese, ambiente de simulação e compartilhamento de informação entre pesquisadores. O ambiente de simulação será formado por *plugins* que encapsulará técnicas de inferências, tais como redes neurais artificiais.

O LIMS (*Laboratory Information Management Systems*) é um dos tipos de sistemas que pode ser usado para esta finalidade. O LIMS é desenvolvido com o objetivo de facilitar o gerenciamento e armazenamento de dados de laboratório, contemplando todo o ciclo de vida dos dados (coleta de dados, armazenamento, análise, geração de relatórios e arquivamento), permitindo e controlando o acesso dos usuários aos resultados armazenados [Hinton, 1995].

Devido à complexidade dos processos gerenciados, o LIMS costuma ser específico para cada laboratório, sendo difícil o uso de um mesmo LIMS por laboratórios diferentes devido às diferenças nos processos de cada um. Assim, a integração de laboratórios diferentes com um mesmo sistema de gerenciamento é um dos desafios da utilização destes. Portanto, é essencial o desenvolvimento de LIMS customizado para a gerência de processos científicos. Entre estes processos está a fabricação de compostos nanométricos. Através da utilização dos LIMS, alguns dos principais problemas dos laboratórios de nanotecnologia atuais, como a falta e/ou precariedade de armazenamento de dados, serão resolvidos.

Um sistema de inferência de dados é uma alternativa para auxiliar o pesquisador na descoberta de conhecimento. Já existem na literatura técnicas propostas para realizar a inferência de dados, como a *Programação Genética* – subárea da computação evolutiva que usa analogia à teoria da evolução de Darwin – e outros baseados em *Redes Neurais Artificiais*, técnicas computacionais inspirados na estrutura neural de organismos inteligentes.

O ambiente de simulação proposto é constituído por sistemas de inferências de

dados, estes são encapsulados em módulos complementares e incorporados à estrutura do LIMS como *plugins*. O encapsulamento de cada simulador como *plugin* permitirá a evolução de cada sistema de inferência independentemente da evolução do LIMS, além de permitir que outros simuladores sejam integrados à ferramenta posteriormente.

O uso de sistemas computacionais para gerenciar dados de fabricação de nanoestruturas provê uma plataforma de armazenamento com permissão de acesso aos dados para os usuários autorizados e protegendo os mesmos de acessos não autorizados. Com a construção deste sistema é possível uma melhora na qualidade e velocidade em que os dados são armazenados. O uso de uma ferramenta automática possibilita isto, uma vez que qualquer dado inserido irá passar por um processo de validação automático e a análise dos dados inseridos também pode ser realizada. O *NanoTrack* supre as principais necessidade de gerenciamento de dados de laboratórios de nanotecnologia, provendo uma ferramenta eficiente e segura para a utilização por pesquisadores da área.

### 1.3 Resumo dos resultados

Neste trabalho foram desenvolvidos três estudos de casos de laboratórios de nanotecnologia: Pontos Quânticos, LabSem - Laboratório de Semicondutores (PUC-Rio) e Nanotubos de Carbono (Física-UFMG). Os processos existentes em cada laboratório foram identificados e organizados em forma de um *workflow*, em seguida foram incorporados ao *NanoTrack*. Além disto, foram desenvolvidos sistemas de inferência para os dois primeiros estudos de casos e estes foram encapsulados em *plugins* e incorporados aos respectivos *workflows*.

Para o primeiro estudo de caso, foi utilizado apenas Redes Neurais Artificiais e o simulador é capaz de obter valores de densidade, altura média e desvio padrão dos pontos quânticos com base em um conjunto de entradas de síntese. Para evitar que o simulador perca a precisão à medida em que novos experimentos são realizados e inseridos no sistema, foi desenvolvido também um *plugin* capaz de realizar o treinamento da rede neural que deverá ser executado sempre que a Rede Neural perder precisão em seus resultados. A saída da rede neural para densidade e altura média obteve erros pequenos, a saída de desvio padrão obteve erro relativamente alto.

No segundo estudo de caso, um simulador foi desenvolvido para obter os parâmetros de síntese de quaternários. Com base nas concentrações dos elementos *Gálio*, *Alumínio* e *Índio* desejadas pelo pesquisador, um algoritmo genético otimiza os parâmetros de síntese. A validação da qualidade dos compostos gerados (indivíduos) é realizada por uma Rede Neural que irá determinar as concentrações. Quanto menor a

diferença entre a concentração desejada e a concentração obtida pela Rede Neural, melhor é o indivíduo. O simulador apresentou resultado satisfatório com um erro bastante pequeno.

O terceiro estudo de caso, *Nanotubos de Carbono*, foi a organização do *Laboratório de Nanomateriais* em forma de um *workflow* e implementação deste na ferramenta *NanoTrack*. Devido a carência de dados de experimentos, não foi possível criar um simulador para este laboratório. Entretanto, através da utilização do *NanoTrack*, será possível desenvolver um algoritmo de inferência e este poderá ser incorporado à ferramenta no futuro.

## 1.4 Contribuições

O desenvolvimento deste trabalho traz pelo menos três grandes contribuições, a primeira e principal contribuição deste trabalho é o desenvolvimento de um sistema computacional, *NanoTrack*, capaz de suprir uma das principais carências encontradas hoje nos laboratórios de nanotecnologia. Com a grande flexibilidade de mapeamento de *workflows* provida pelo sistema **Flux®**, plataforma base do *NanoTrack*, foi possível criar três instâncias distintas de laboratórios de nanotecnologia: **Pontos Quânticos, Laboratório de Semicondutores (Puc-Rio) e Nanotubos de Carbono (Física-UFMG)**.

A segunda contribuição deste trabalho, que é uma consequência da anterior, são os algoritmos de inferência que foram utilizados nos *plugins* desenvolvidos. Eles apresentaram ótimos resultados com uma margem de erro bastante pequena, o que permite maior confiabilidade e demonstra maturidade da idéia proposta.

A terceira contribuição foram as cooperações com os laboratórios de pesquisas, tanto os laboratórios que foram beneficiados com o desenvolvimento dos sistemas computacionais para gerência de suas informações como também a parceria com o laboratório *LUAR (Laboratório de Universalização do Acesso - DCC/UFMG)* que através das interações possibilitou um avanço no desenvolvimento da plataforma base **Flux®**.

Além destas contribuições, o desenvolvimento deste trabalho abre oportunidades para outros laboratórios de nanotecnologia interessados a incorporar o *NanoTrack* como ferramenta principal para gerência de suas informações e, conseqüentemente, facilitar a descoberta de conhecimento para outros pesquisadores. Abre também uma grande diversidade de possibilidades para o desenvolvimento de trabalhos futuros para outros alunos de graduação, mestrado e doutorado.

## 1.5 Organização da dissertação

Este trabalho está organizado em seis capítulos. Este primeiro capítulo faz uma introdução onde é apresentado o problema abordado, a motivação do trabalho e suas contribuições. O capítulo 2 apresenta conceitos fundamentais para a compreensão do trabalho, tais como: escala nano, LIMS, *workflow*, redes neurais artificiais e nanoinformática. O capítulo 3 mostra a revisão bibliográfica apresentando os principais trabalhos relacionados. No capítulo 4 é apresentado o *NanoTrack*, a modelagem e a integração dos *plugins* com o LIMS, mostrando as vantagens e os benefícios que este trabalho trará para os pesquisadores. O capítulo 5 mostra os três estudos de caso realizados durante o projeto. Finalmente no sexto capítulo são apresentadas as conclusões e propostos trabalhos futuros.



# Capítulo 2

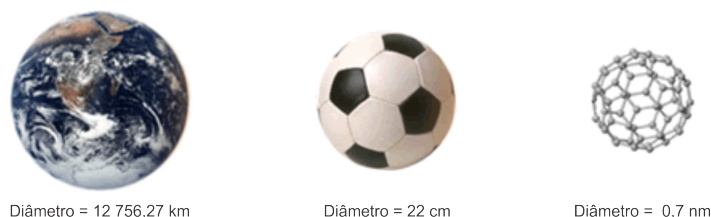
## Fundamentos

### 2.1 A escala nano

Os prefixos centi, mili, micro e nano, por exemplo, são usados para que se possa especificar o fator pelo qual é multiplicada uma determinada grandeza. Estes fatores estão associados com potências de 10. Assim, como o kilo corresponde a um fator  $10^3$ , mili corresponde a  $10^{-3}$ . Na escala nano, o fator corresponde a  $10^{-9}$ . Desta forma, quando se fala de 1 nanômetro (nm), refere-se a um fator de  $10^{-9}$  do metro, um bilionésio do metro. É nessa escala de tamanho que a nanotecnologia é trabalhada e que os objetos nanotecnológicos são concebidos, a mesma escala dos átomos e moléculas [Vilela Neto & Pacheco, 2012].

Para se ter uma idéia de quão pequeno é a medida nanométrica, a imagem mostrada na Figura 2.1 realça uma relação aproximadamente equivalente entre o planeta Terra, uma bola de futebol e uma estrutura nanométrica, neste caso um fullereno.

A Figura 2.2 mostra alguns elementos cujo tamanhos são inferiores a 1 metro, com destaque para o lado esquerdo da imagem que demonstra uma escala nanométrica e é nesta escala que estão localizadas partículas como o dióxido de titânio/platina, a fita de DNA, os nanotubos de carbono dentre várias outras estruturas. A escala exibida



**Figura 2.1.** Relação equivalente de escalas.

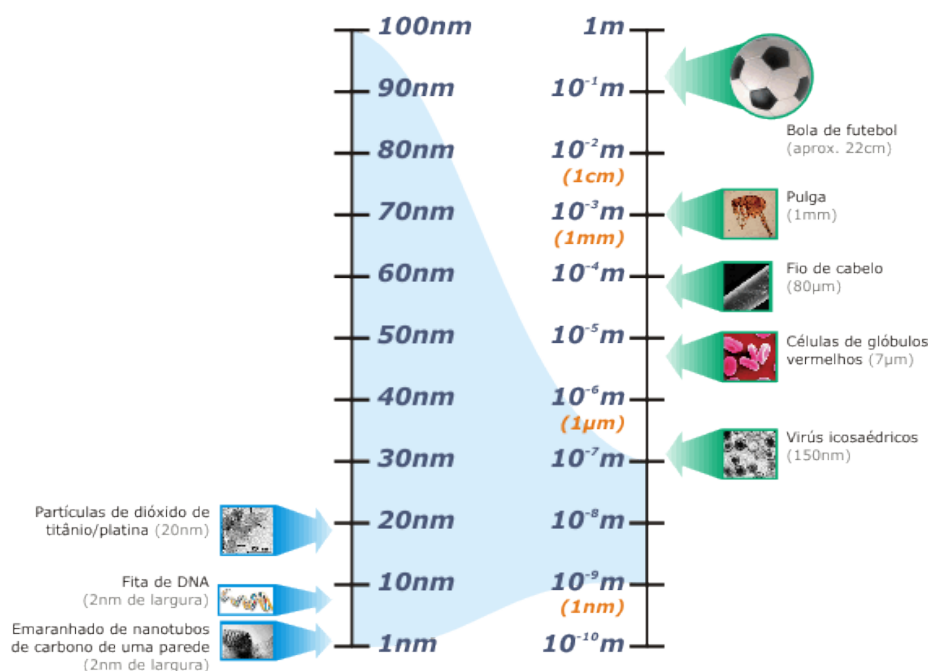


Figura 2.2. Escalas abaixo de 1 metro.

no lado direito da imagem é um demonstrativo do quão pequeno são as estruturas nanométricas com relação aos demais elementos.

Trabalhar na escala nanométrica não é uma tarefa trivial e, por isto, cria novos grandes desafios à ciência. Estes desafios já vêm sendo estudados há alguns anos, mas por ser um campo de estudo ainda recente, existe muito a ser desvendado. Nesta escala a física revela algumas características peculiares e que são pouco conhecidas.

## 2.2 Sistemas de Gerenciamento de Informação de Laboratórios - LIMS

Conforme mostrado em [Bell et al., 2009], algumas áreas da ciência estão passando por um momento de aumento significativo de quantidade de dados gerados. Seguindo esta tendência, os laboratórios dependem cada vez mais de sistemas computacionais capazes de gerir os dados gerados em seu dia-a-dia. Os equipamentos de alta tecnologia encontrados nestes laboratórios normalmente possuem softwares integrados capazes de auxiliar o gerenciamento dos dados gerados pelo equipamento, entretanto este gerenciamento é descentralizado onde cada equipamento controla os dados gerados por ele próprio. Contudo, é muito importante para os laboratórios uma análise global dos seus dados [Melo et al., 2010] e um LIMS é capaz de centralizar estas informações em um

## 2.2. SISTEMAS DE GERENCIAMENTO DE INFORMAÇÃO DE LABORATÓRIOS - LIMS

único ambiente.

O LIMS (*Laboratory Information Management Systems*) é desenvolvido com o objetivo de facilitar o gerenciamento e armazenamento de dados de laboratórios, contemplando todo o ciclo de vida dos dados (coleta de dados, armazenamento, análise, geração de relatórios e arquivamento), permitindo o acesso dos usuários aos respectivos resultados armazenados [Hinton, 1995]. Através da utilização dos LIMS, os laboratórios conseguem reorganizar melhor sua organização interna tratando os pontos críticos de sua estrutura e proporcionando diversos benefícios. Além disso, um LIMS dá ênfase na melhoria de qualidade dos dados e busca gerar seus resultados de maneira consistente e confiável [Melo et al., 2010].

Para que um sistema seja considerado um LIMS alguns requisitos básicos devem ser observados, tais como:

- Armazenamento e rastreabilidade de amostras;
- Consulta de informação e geração de relatórios;
- Gerenciamento de atividades;
- Gestão de qualidade (auditoria, documentação, não-conformidades, etc.);
- Calibração e manutenção dos instrumentos dos laboratórios;
- Sistema de login a fim de restringir o acesso aos dados do LIMS;
- Integração com outros sistemas e equipamentos;
- Gerenciamento de materiais de insumos.

Um LIMS, apesar de estar suscetível à mudanças, tende a ter um conjunto básico de funcionalidades que pode ser dividido em cinco fases de processamento do experimento no laboratório [Skobelev et al., 2011]:

- Obtenção e registro da amostra;
- Atribuição, programação e acompanhamento dos procedimentos que serão aplicados à amostra;
- Processamento e controle de qualidade da amostra e dos equipamentos utilizados;
- Armazenamento dos dados de análise da amostra;

- Inspeção, aprovação e compilação dos dados da amostra para gerar relatórios ou outras análises.

O LIMS deve integrar todas as atividades do laboratório, possibilitando o gerenciamento de todo o laboratório apenas com o LIMS [Melo et al., 2010]. Além disto, o LIMS contribui para a garantia da qualidade dos procedimentos, a integração com outros laboratórios, garante a capacidade de manipulação de dados e também permite gerar relatórios e gráficos para análises.

### 2.2.1 Gerenciamento da explosão da informação

A preocupação com o grande volume de dados que estão sendo gerados é cada vez maior, uma vez que estes são de extrema importância tanto para fins de registro quanto para a descoberta de conhecimento. Para que os dados gerados durante as atividades de pesquisa do laboratório não sejam perdidos é necessário que o sistema dê suporte ao usuário para que ele possa não apenas adicionar os dados no sistema como também permitir que ele possua maneiras eficientes de consultar estes dados em um momento futuro. O LIMS possui funcionalidades que visam atender esta necessidade, garantindo a organização e armazenamento destes dados, além de realizar uma interação com os diversos equipamentos do laboratório, o que permite a inserção direta do grande volume de dados gerados pelos equipamentos em sua base de dados, além de permitir que os dados armazenados no sistema sejam exportados para outras ferramentas.

### 2.2.2 Garantia da qualidade

A preocupação com a qualidade já vem sendo alvo de estudos há bastante tempo e com isto os conceitos de qualidade sofreram uma evolução considerável em função das exigências mercadológicas [Melo et al., 2010]. A aplicação do controle de qualidade é essencial para o desenvolvimento de qualquer entidade, uma vez que sem o controle é difícil detectar falhas nos procedimentos, o que impede que estes sejam continuamente melhorados. Além disto, o governo vem criando uma série de regulamentos, que a cada dia estão mais rigorosos, e que visam melhorar os processos de qualidades.

Para implantar um sistema de qualidade em um laboratório é necessário criar procedimentos adequados para todas as suas atividades, com base nas orientações do sistema de qualidade que foi adotado pelo laboratório. Existem algumas organizações responsáveis por assegurar a garantia da qualidade [Botelho, 2012]:

- ISO – International Organization for Standardization – É uma organização não-governamental responsável por elaborar padrões e normas internacionais a fim

## 2.2. SISTEMAS DE GERENCIAMENTO DE INFORMAÇÃO DE LABORATÓRIOS - LIMS

de facilitar as relações comerciais entre países. No Brasil ela é representada pela ABNT (Associação Brasileira de Normas Técnicas);

- INMETRO – Instituto Nacional de Metrologia, Qualidade e Tecnologia – Autarquia federal - vinculada ao Ministério do Desenvolvimento, Indústria e Comércio Exterior - que apresenta, entre outras atividades, a responsabilidade em credenciar laboratórios que voluntariamente optarem por esta decisão.

Além destas organizações, existem sistemas de qualidades, tal como BPL – Boas Práticas de Laboratório, específicos que podem ser adotados pelo laboratório, são sistemas relativos ao processo organizacional e às condições sob as quais estudos de laboratórios são planejados, executados, monitorados, registrados e relatados [Inmetro, 2012].

Com o uso de sistemas computacionais, o atendimento aos requisitos das orientações do sistema de qualidade é facilitado, uma vez que este traz diversas contribuições positivas para a garantia da qualidade do laboratório. Veja a seguir algumas das contribuições que podem ser alcançadas através da utilização de LIMS:

- Redução de erros na entrada de dados;
- Automatização de validações dos dados;
- Permite auditar os dados gerados;
- Buscas por dados mais eficientes;
- Documentação detalhada das atividades;
- Geração de relatórios.

### 2.2.3 Maior eficiência e eficácia na execução de atividades do laboratório

Através da utilização de LIMS é possível alcançar maior eficiência na realização das atividades do laboratório, uma vez que os procedimentos estarão padronizados e terá um ambiente propício à realização das atividades com maior produtividade. Consequentemente, as decisões sobre o que deverá ser feito a cada momento também serão mais precisas, alcançando a maior eficácia no laboratório.

O LIMS implementa diversas funcionalidades que possibilitam melhorar a eficiência e a eficácia, são elas:

- Cálculos automáticos;
- Geração automática de relatórios;
- Validação de dados;
- Entrada automática de dados;
- Consulta à dados antigos;
- Integração com ferramentas de terceiros.

## 2.3 Workflow

Em 1993 foi fundado um consórcio formado por grandes empresas para definir padrões de interoperabilidade de sistemas de gestão de *workflow*, a *Workflow Management Coalition (WfMC)*. Segundo [Hollingsworth et al., 1995] e [WfMC, 2012], *workflow* consiste em uma sequência de passos necessários para que se possa atingir a automação de processos de um procedimento, de acordo com um conjunto de regras definidas, envolvendo a noção de processos, regras de operação e controle de dados, permitindo que este conjunto de informações possa ser transmitido de uma pessoa para outra sem sofrer alterações e de maneira clara e concisa.

As atividades de um *workflow* podem ocorrer de forma concorrente e eventualmente impactar umas nas outras, de acordo com um conjunto de regras. A automação do processo de negócio identifica as várias atividades do processo, regras de procedimentos e controle de dados associados para gerenciar o *workflow* durante a ativação do processo. Muitas instâncias individuais fazem parte de um mesmo processo. Todas devem atuar de forma harmônica para se tornarem operacionais, cada uma associada a um conjunto específico de dados relevantes.

### 2.3.1 XPDL - XML Process Definition Language

O *Workflow Management Coalition (WfMC)* propõe um padrão de uma linguagem para definição de processos, o *XML Process Definition Language - XPDL*. Isto porque existe uma grande variedade de ferramentas que podem ser usadas para analisar, modelar, descrever e documentar um processo de negócio. Este padrão proposto é uma interface de definição de processo que é responsável por intermediar a interoperabilidade entre diferentes produtos e ferramentas de fluxos de trabalho [Interface, 2002]. O *XPDL* utiliza a notação XML para realizar a especificação do fluxo de trabalho.

O *XPDL* especifica conceitos importantes que devem ser atendidos no momento da definição de processos de negócio, tanto em aspecto técnico quanto no aspecto de informações gráficas. Tais orientações são compatíveis com o padrão *BPMN - Business Process Modeling Notation*, que é o padrão para modelar o processo em aspectos visuais, entretanto os conceitos presentes em *XPDL* são abordados de maneira mais técnica.

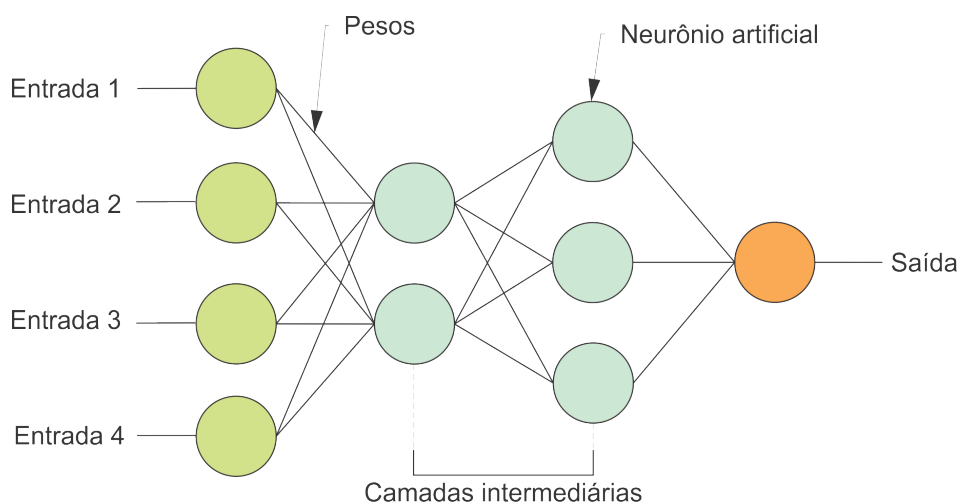
O fluxo de trabalho é representado no *XPDL* através de objetos, relacionamentos e atributos. Cada objeto pode possuir um conjunto de atributos que irão caracterizá-lo e um conjunto de ligações a outros objetos que irá formar os relacionamentos. Os principais elementos para a definição de fluxo de trabalho são os processos e as atividades que possuem características definidas como atributos e transições. Além de processos e atividades, o *XPDL* é constituído por outros elementos como: transição, aplicação, variáveis, atributos estendidos, parâmetros formais, grupo de atividades, participantes, aplicações e pacotes.

## 2.4 Redes Neurais Artificiais

Redes Neurais Artificiais, ou simplesmente redes neurais, representam uma tecnologia que tem raízes em muitas disciplinas: neurociência, matemática, estatística, ciência da computação e engenharia. As redes neurais encontram aplicações em campos tão diversos, como modelagem, análise de séries temporais, reconhecimento de padrões, processamento de sinais e controle, em virtude de uma importante propriedade: a habilidade de aprender a partir de dados de entrada com ou sem um professor [Haykin, 2001].

A idéia de desenvolver neurocomputação surgiu através da observação do cérebro humano que possui características extremamente avançadas e pode ser comparado a um computador de alto desempenho, capaz de realizar processamento paralelo, ser altamente complexo e não-linear. Os primeiros trabalhos que surgiram foram propostos no trabalho [Pitts & McCulloch, 1947] onde os autores modelaram o processo eletrônico baseado em neurônios biológicos e o valor da saída era o resultado baseado em uma função que tratava as diversas entradas.

Desde então, muitos trabalhos foram desenvolvidos, o que proporcionou um processo de amadurecimento das técnicas de neurocomputação bem como a criação de novos modelos de redes neurais como as redes *Perceptrons* proposta por [Rosenblatt, 1962] e que mais tarde foram aprimoradas com um algoritmo de retropropagação do erro, denominado de *Backpropagation* [Haykin, 2001], cuja descoberta revolucionou a área de redes neurais.



**Figura 2.3.** Exemplo de uma rede neural simples com duas camadas intermediárias.

Segundo [Vilela Neto & Pacheco, 2012], a estrutura de uma rede neural é composta por três elementos básicos que são: o modelo do neurônio artificial, a estrutura de ligações entre os neurônios e a regra de aprendizado. Os nós interligados na rede são os neurônios artificiais, responsáveis pelo processamento e são altamente interconectados. Eles realizam operações simples, transmitindo seus resultados aos neurônios vizinhos. As arestas entre os nós armazenam funções de peso que são responsáveis por realizar ajustes que determinará o efeito da entrada sobre o neurônio. A estrutura básica de uma rede neural é apresentada na Figura 2.3.

A grande capacidade da rede neural em extrair conhecimento através de exemplos e conseguir generalizar as informações é um dos principais fatores para a utilização de redes neurais na solução de problemas. Além disto, a habilidade das redes neurais em realizar mapeamentos não lineares entre suas entradas e saídas as têm tornado prósperas no reconhecimento de padrões e na modelagem de sistemas complexos [Vilela Neto & Pacheco, 2012].

A generalização é a capacidade da rede neural aprender através de um conjunto limitado de exemplos e conseguir apresentar respostas coerentes para dados desconhecidos. Com um bom tratamento de dados, para evitar que haja ruídos nas amostras utilizadas durante o processo de treinamento da rede neural, e através da generalização é possível fazer descoberta de conhecimento muito além do que simplesmente mapear relações entre entradas e saídas.

Dado um conjunto de entrada e saídas conhecidos, o processo de aprendizado de uma rede neural possui algumas etapas básicas: tratamento de dados, seleção de amostras e o treinamento.



### 2.4.1 Tratamento de dados

A etapa de tratamento de dados visa eliminar possíveis ruídos do conjunto de amostras. Ela é importante porque se a rede for treinada com amostra cujas informações não são precisas o resultado final obtido será uma rede neural mal calibrada, desta forma ela não irá conseguir alcançar bons resultados.

Eliminar ruídos do conjunto de amostra pode não ser uma tarefa fácil. Além do processo de separar amostras ruins ser um procedimento trabalhoso, ela depende muito do conhecimento do pesquisador em saber distinguir uma amostra boa de uma amostra ruim.

É comum que os pesquisadores, a cada experimento prático realizado cujo resultado está um pouco divergente do que era esperado, marque a amostra como uma amostra duvidosa. Isto porque pode ter ocorrido algum evento durante o processo de experimentação que ocasionou as divergências de resultados. E estes eventos normalmente são anotados para uma futura identificação de anormalidade.

### 2.4.2 Seleção de amostras

A seleção das amostras é uma importante parte do processo de treinamento e pode ser determinante para o bom aprendizado da rede neural. Nesta etapa os dados que serão utilizados durante o treinamento são selecionados e é importante que haja uma boa diversidade de amostras para que a rede neural não se especialize no problema e consiga generalizar.

### 2.4.3 Treinamento

O treinamento é a principal etapa do processo e o objetivo dele é conseguir ajustar os pesos das ligações dos neurônios artificiais a fim de reduzir os erros obtidos com o resultado de saída da rede. Existem alguns tipos de algoritmos de aprendizagem que implementam técnicas para realizar estes ajustes, mas duas destas técnicas se destacam que são os algoritmos de aprendizagem supervisionado e não-supervisionado.

- Supervisionado: é apresentado um conjunto de dados para treinamento, com entradas e saídas conhecidas. As saídas são os resultados considerados ótimos para o qual a rede deverá se aproximar. Os valores de entrada serão utilizadas pela rede para que ela possa calcular o valor da saída. Dado o valor da saída, a diferença entre o valor obtido e o valor esperado é considerado o erro. Com base no erro obtido pela rede, os pesos da rede (ligações entre os neurônios artificiais) são reajustados para que se possa tentar minimizá-lo;

- Não-supervisionado: a rede atualiza seus pesos sem o uso de pares entrada-saídas desejadas e sem indicações sobre a adequação das saídas produzidas. Neste tipo de algoritmo é feita uma análise estatística para detectar similaridades entre os padrões de saídas. E a regra de aprendizado adotado é a de aprendizagem competitiva, onde normalmente o melhor neurônio é selecionado para a rede neural.

## 2.5 Nanoinformática

Informática é a aplicação de métodos da ciência da informação e da computação para coletar, analisar e aplicar informações. A “X-informática” tem se tornado o descritor padrão para a aplicação de tais métodos para um conjunto de problemas em um campo ou disciplina específica, tais como bioinformática em biologia ou ecoinformática em ecologia [Hey, 2010]. De maneira similar, “X Computacional” se refere ao uso de métodos computacionais em um campo ou disciplina específica (p. ex.: Astronomia Computacional ou Geologia Computacional). Logo, Nanoinformática é uma metodologia sistemática para automatizar, coletar, organizar, validar, armazenar, compartilhar, analisar e aplicar dados envolvendo Nanociência e Nanotecnologia [de la Iglesia et al., 2011]. Já a Nanociência ou Nanotecnologia Computacional inclui o desenvolvimento de aplicação de ferramentas críticas necessárias para a simulação, computação e previsão de modelos de nanomateriais, nanodispositivos e nanosistemas.

A Nanoinformática é necessária para o desenvolvimento inteligente e caracterização comparativa de nanomateriais, para o projeto e uso de nanodispositivos e nanosistemas otimizados, para o desenvolvimento de instrumentação avançada e processos de manufatura e para garantir a segurança ambiental e da saúde dos envolvidos. A Nanoinformática também fomenta a descoberta científica eficiente através das técnicas de mineração de dados, aprendizado de máquina e otimização. Ela também envolve o uso de ferramentas de rede de comunicação para a eficiente troca e disponibilização da informação importante de forma eficiente.

A Figura 2.4, retirada do livro “The Fourth Paradigm: Data-Intensive Scientific Discovery” [Hey, 2009] ilustra bem esta questão. Os experimentos, a literatura, as simulações e outros arquivos nos enchem de informações. Os cientistas desejam codificar estas informações de modo que as respostas às suas dúvidas sejam supridas e que eles possam compartilhar de maneira eficiente com outros cientistas. A maneira mais rápida, produtiva e eficiente para que o pesquisador possa entender as informações armazenadas em seu computador é se os próprios programas nele instalados também

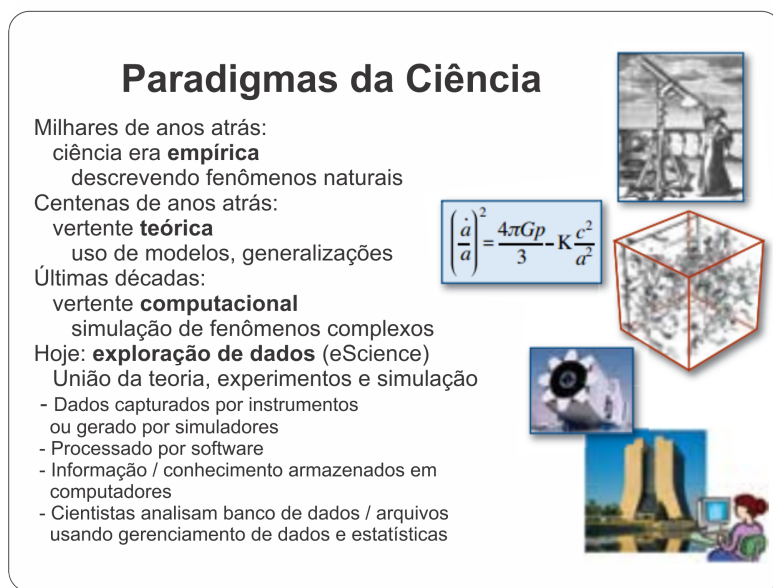


**Figura 2.4.** X-Informática.

reconhecerem as informações. Ou seja, a informação tem que ser representada de uma forma algorítmica.

Nas últimas duas décadas, a exploração de dados em larga escala começou combinando experimentos que geram dados e Ciência da Computação, usando redes maciças de computadores, ferramenta de ciência da informação e, até mesmo, tecnologias de redes sociais. Estes projetos precederam o início da era da *e-Science* (ou e-Ciência, em português) [Hey, 2009]. Exemplos populares de e-Science são o Projeto do Genoma Humano [of Energy Office of Science, 2012] e o “Sloan Digital Sky Survey” [Kent, 1994]. Estes projetos demonstram como sistemas computacionais sofisticados e a coordenação do domínio do especialista podem se harmonizar para resolver grandes desafios científicos. A Nanoinformática - a aplicação do paradigma de e-Science para a ciência em nanoescala - objetiva a solução de desafios para a aplicação da Nanotecnologia em benefício da sociedade. Contudo, é importante salientar que a área de Nanociência e a Nanotecnologia tem suas próprias particularidades, sendo a mais importante delas, seu caráter fortemente multidisciplinar. Logo, o desenvolvimento de ferramentas que atendam atores com vocabulários distintos é um dos grandes desafios enfrentados no desenvolvimento da Nanoinformática.

A Figura 2.5, também retirada de [Hey, 2009], mostra que o mundo da ciência mudou muito ao longo dos anos. Originalmente havia apenas a ciência experimental. Em seguida surgiu a ciência teórica com as leis de Kepler, as leis de Newton, as equações de Maxwell e assim por diante. Então, para muitos problemas, os modelos teóricos



**Figura 2.5.** Paradigmas da ciência.

cresceram em demasia para serem resolvidos de forma analítica e então os cientistas começaram a simular. No momento atual, estas simulações tem gerado uma grande quantidade de dados, somados ao enorme crescimento da geração de dados produzidos pelas ciências experimentais. O e-Science (o quarto paradigma) surge justamente para unificar a teoria, o experimento e a simulação.

Especificamente no caso da Nanociência e Nanotecnologia, os dados tem sido gerados continuamente por softwares de modelagem molecular (Gaussian, Gammes, Siesta, etc), simulação de nanodispositivos (SimOled, MEEP, QCADesigner, etc), novos modelos baseados nas teorias da mecânica quântica, ferramentas de síntese e crescimento (CVD, FIB, Nanolitografia, etc) e ferramentas de caracterização (Raio-X, Ramam, RMN, Microscopias, etc). Esta grande quantidade de dados deve ser armazenada e manipulada de forma eficiente de modo que as descobertas precisas possam ser feitas e as informações compartilhadas.

A Nanoinformática faz uma ligação direta entre o desenvolvimento experimental e a teoria. Primeiramente ela está fortemente relacionada ao experimento, automatizando o processo, coletando e armazenando os dados por ele gerado. Em um segundo momento a Nanoinformática passa a ter uma forte relação com a teoria, já que os dados gerados pela parte experimental podem ser usados para o desenvolvimento de novos modelos teóricos, sistemas de previsão de síntese, modelos de mineração de dados para a descoberta de conhecimento, simulações e otimizações de processos e de geometria dos nanomateriais e nanodispositivos. Por fim, as informações geradas pela segunda

etapa podem ser utilizadas para a posterior produção de nanoestruturas otimizadas e aprimoradas, ou seja, retornando à fase experimental.



# Capítulo 3

## Trabalhos relacionados

Para que haja evoluções na ciência é muito importante que o “estado da arte” seja conhecido, caso contrário, pesquisadores irão investir seu tempo em trabalhos que poderiam ser menos complicados se eles tivessem acesso a outros meios de auxílio já conhecidos pela academia. Novas descobertas envolvem combinações de resultados de pesquisas já realizadas e, além disto, a idéia de raciocínio comparativo também pode fornecer meios para prever resultados similares às já conhecidas e experimentos computacionais complementam os experimentos realizados em laboratórios.

A biologia, por exemplo, foi uma das primeiras áreas da ciência a ser auxiliada por sistemas computacionais, dando origem à bioinformática. Para que a recém criada área de nanoinformática possa evoluir é necessário conhecer os principais erros e acertos que as outras áreas que utilizam os sistemas computacionais tiveram e utilizar esta informação para dar suporte à nanoinformática.

As ferramentas de software são desenvolvidas para solucionar problemas. Normalmente elas implementam alguma técnica conhecida que é comprovadamente capaz de resolver o problema, mas existe a dificuldade de como a ferramenta será evoluída na medida em que técnicas alternativas vão surgindo. Em engenharia de software são realizados diversos estudos sobre métodos de desenvolvimento de software que objetivam facilitar tanto a manutenção quanto a evolução de um software como discutido em [Bennett & Rajlich, 2000].

### 3.1 LIMS e Workflows

Já existem vários algoritmos e ferramentas computacionais para gerenciamento de dados de laboratório, mas sistemas destinados ao uso de laboratórios de nanotecnologia são raros e pouco flexíveis [de la Iglesia et al., 2011]. Diante do avanço constante e

rápido da área, projetar um software que seja capaz de resistir a estes avanços constitui um desafio. Dois trabalhos apresentados na literatura [Wermelinger et al., 2008] e [Crnkovic et al., 2002] exploraram os princípios dos projetos conhecidos por impactar na manutenção de software e planejaram um modelo de modularização de ferramenta baseado nestes princípios para permitir a evolução da ferramenta através de *plugins* simples e ajustáveis.

O LIMS já vem sendo adotado por muitos laboratórios, mas, normalmente, estes sistemas são comerciais ou foram desenvolvidos para um laboratório específico, sendo que sua estrutura é pouco flexível à alterações. Além disto, possuem código fonte fechado e um alto custo. Core [Informatics, 2013], [Prog4biz, 2013] e Sql [Lims, 2013] são algumas empresas que comercializam softwares para gerenciamento de informações de laboratórios, sendo que as duas primeiras desenvolvem LIMS comerciais para nanotecnologia. De acordo com as especificações disponíveis no acervo das empresas, estes LIMS atendem os requisitos abordados na seção 2.2, entretanto nenhum destes LIMS integra nenhum tipo de ferramenta de simulação.

[Ludäscher et al., 2006], criador do Sistema Kepler, e [Oinn et al., 2006], criador do Taverna, descrevem ferramentas que contribuem para realização de pesquisas científicas na área da bioinformática, biologia, ecologia e geologia. O projeto de pesquisa é modelado em forma de um *workflow* onde as atividades são alocadas segundo uma ordem definida pelo pesquisador e o resultado obtido em uma atividade serve como entrada para a atividade seguinte. O conjunto dos resultados do fluxo de atividade forma a solução do problema.

O Taverna [Oinn et al., 2006], interliga diversos serviços web (*webservices*) dedicados. Cada serviço é desenvolvido por terceiros de acordo com a necessidade dos próprios pesquisadores e são disponibilizados em servidores na internet. Estes serviços são incorporados ao *workflow* como *plugins*. O *workflow* é construído baseado em uma linguagem própria chamada de *SCUFL - Simple Conceptual Unified Flow Language*.

Estas ferramentas contribuem para o compartilhamento das pesquisas realizadas, assim, um pesquisador consegue ter acesso à informações de pesquisa de um outro pesquisador, contribuindo para o avanço das pesquisas. Em nanotecnologia ainda não existem ferramentas não-comerciais com este propósito, sendo as raras ferramentas computacionais baseadas em *scripts*, que são especializados no problema tratado e pouco flexíveis.

Além disso, a nanociência e a nanotecnologia apresentam um caráter multidisciplinar, o que torna o desafio ainda maior, já que neste caso cientistas de diferentes áreas (física, química, biologia, engenharia, etc) muitas vezes desejam compartilhar as mesmas informações, contudo cada área tem seus padrões específicos. Logo, o desafio



de desenvolver ferramentas flexíveis e funcionais é grande, como mostra [de la Iglesia et al., 2011].

As necessidades de gerenciamento de dados de laboratórios vem crescendo a cada ano. Laboratórios de pesquisas estão cada vez mais preocupados em não perder nenhuma informação e com isto a demanda por software capazes de fazer este controle, LIMS, também vem crescendo. Pensando nas dificuldades encontradas na criação de softwares de modo a atender as diversas áreas de pesquisa existente, [Melo et al., 2010] propôs um software amplamente flexível, o SIGLa, *Sistema Integrado de Gerência de Laboratório*.

O Flux<sup>®</sup>, uma extensão do sistema SIGLa [Melo et al., 2010], é um LIMS com código aberto e capaz de adaptar suas atividades e processos para diversos tipos de laboratórios. Ele possui um sistema de gerenciamento de *workflows* integrado ao seu código, desta forma, é possível criar e gerenciar diversos *workflows* personalizados. Para cada laboratório, suas atividades e processos são mapeados e descritos em um arquivo XML juntamente com os seus atributos. O arquivo XML pode ser gerado através dos editores visuais *Toghter Workflow Editor* e *Fluxar*, tornando este procedimento mais simples e eficaz. Nele, o núcleo de processamento de informações não é acoplado ao *workflow* de um laboratório específico. Portanto, se alguma atividade do laboratório for criada, modificada ou removida, basta modificar a estrutura do *workflow* para que as modificações sejam reconhecidas pela ferramenta.

## 3.2 Inferência de nanoestruturas

Para auxiliar os especialistas na síntese de futuras nanoestruturas, este trabalho propõe uma ferramenta com sistemas inteligentes de inferência e previsão integrados ao *workflow* como *plugins* que permitirão a aplicação de técnicas específicas de inferência quando necessário. A abordagem orientada a *workflows* do *NanoTrack* permite que o processo e a aplicação evoluam paralelamente e de maneira independente, pois a lógica do processo está totalmente separada da lógica da aplicação. Isto também provê uma grande flexibilidade, pois, caso o processo de fabricação mude ou haja a necessidade de se armazenar qualquer outro dado a única coisa que precisa ser alterada é o *workflow*, e não o código-fonte. Além disso, essa abordagem, baseada em *plugins*, irá possibilitar que a ferramenta acompanhe a evolução das técnicas e facilitará a introdução de novas alternativas.

Encapsular as técnicas de inferência de dados em *plugins*, além de facilitar a evolução da ferramenta, irá permitir a utilização de múltiplas técnicas com a incorporação

de múltiplos *plugins* na mesma ferramenta. Esta possibilidade trará vantagens para o sistema, uma vez que uma única técnica não é boa o suficiente para garantir a melhor solução para todos os tipos de problemas apresentados, e logo, a diversidade de técnicas é interessante.

Cada *plugin* poderá ser desenvolvido utilizando algoritmos diferentes. Hoje em dia é muito comum a utilização de redes neurais artificiais, algoritmos genéticos, programação genética, algoritmos probabilísticos, dentre outros. Com a evolução da computação é provável que as atuais técnicas sejam aperfeiçoadas e novas técnicas surjam. Logo, deixar um espaço para a incorporação de novas técnicas se faz necessário para a constante evolução da ferramenta.

A utilização de algoritmos de inferência já vem sendo aplicado à síntese de nanoestruturas. Existem muitos trabalhos disponíveis na literatura de algoritmos capazes de resolver problemas específicos, tais como [Dashtbayazi et al., 2007], que utilizou redes neurais artificiais para modelar os efeitos de parâmetros mecânicos de fusão de nanocompósitos. Outro estudo que utilizou redes neurais artificiais foi [Bakucz et al., 2008] que desenvolveu um algoritmo para verificar os efeitos dos desgastes das pontas dos microscópios nas imagens de microscopia AFM. Isto é relevante pois, uma vez que o desgaste irá deteriorar a qualidade das imagens, o sistema consegue notificar o pesquisador para que ele troque a ponta do microscópio e assim evitar que as imagens sejam prejudicadas.

[Zhang et al., 2012] integrou em seu trabalho dois métodos de inteligência computacional formando um algoritmo combinado com uma rede neural artificial *backpropagation* e um algoritmo genético que foram utilizados para prever a composição de estruturas de nano e micro compósitos de ferramentas cerâmicas. O algoritmo genético foi utilizado para otimizar alguns parâmetros da rede neural artificial como topologia, limites e pesos das conexões iniciais. A rede neural foi utilizada para inferir as propriedades mecânicas dos materiais, tais como a dureza, resistência à flexão e tenacidade à ruptura. Até então, as composições eram obtidas através de métodos brutos, como tentativa e erro, e com a utilização do algoritmo foi possível obter resultados muito satisfatórios que foram comprovados com a fabricação destes materiais.

Devido as limitações de tempo de processamento computacional necessários durante as simulações de dinâmicas molecular, a busca por alternativas torna-se fundamental. No trabalho apresentado por [Asproulis & Drikakis, 2009] mostra que uma rede neural artificial fez previsões de resultados com boa concordância com os resultados obtidos em simulações reais. Em [Nateri et al., 2011], os autores fazem uma comparação entre dois métodos de simulação de tamanhos de nanopartículas de prata sintetizadas através do processo de *Tollens*. A técnica baseada em regressões polinomiais múltiplas

se mostrou pior diante do algoritmo de rede neural artificial utilizado, sendo que a primeira obteve erros consideravelmente superiores ao do segundo algoritmo.

No trabalho [Singulani et al., 2008] foram implementadas duas técnicas de inteligência computacional, rede neural artificial e algoritmo genético, para prever tendências no comportamento de altura média dos pontos quânticos em InAs em função dos parâmetros de crescimento. Através do treinamento da rede neural artificial utilizada, foi possível, mesmo com uma base de dados pequena, associar os parâmetros de entradas com a altura média do crescimento de pontos quânticos. Com o algoritmo genético, os autores conseguiram obter condições de crescimento de forma a minimizar a altura média dos pontos quânticos. Os resultados obtidos com as técnicas implementadas foram comparados com os resultados experimentais, e os resultados demonstraram que os algoritmos desenvolvidos são bastante eficazes e promissores.

Em [Cupertino et al., 2011] é apresentado um algoritmo de aprendizado baseado em redes neurais artificiais para inferir o módulo de Young (relação entre a deformação e a tensão aplicada em um material) dos nanocompósitos. A criação de novos compostos, em geral, é um procedimento caro. Este trabalho apresenta uma alternativa para predição do comportamento dos compostos realizada por meio computacional, evitando custos de experimentação em busca da melhor combinação de materiais. Os autores conseguiram obter um resultado satisfatório com a rede neural artificial gerada, apresentando grande precisão e baixa percentagem de erros.

Já no trabalho apresentado por [Dias et al., 2010] é apresentado a PGLIQ (Programação Genética Linear com Inspiração Quântica). Este sistema de inferência aproveita os benefícios de desempenho dos algoritmos quânticos para evoluir programas em código de máquina. A estruturação do modelo da ferramenta é baseada no conceito de superposição de estados da física quântica. Segundo [Dias et al., 2010], seu funcionamento é baseado em indivíduos quânticos e eles representam todos os possíveis programas do espaço de busca. No momento em que o indivíduo é observado ele assume características de um indivíduo clássico formando uma solução. Este algoritmo foi utilizado para realizar inferência de dados de pontos quânticos, tais como: altura média, densidade e desvio padrão.

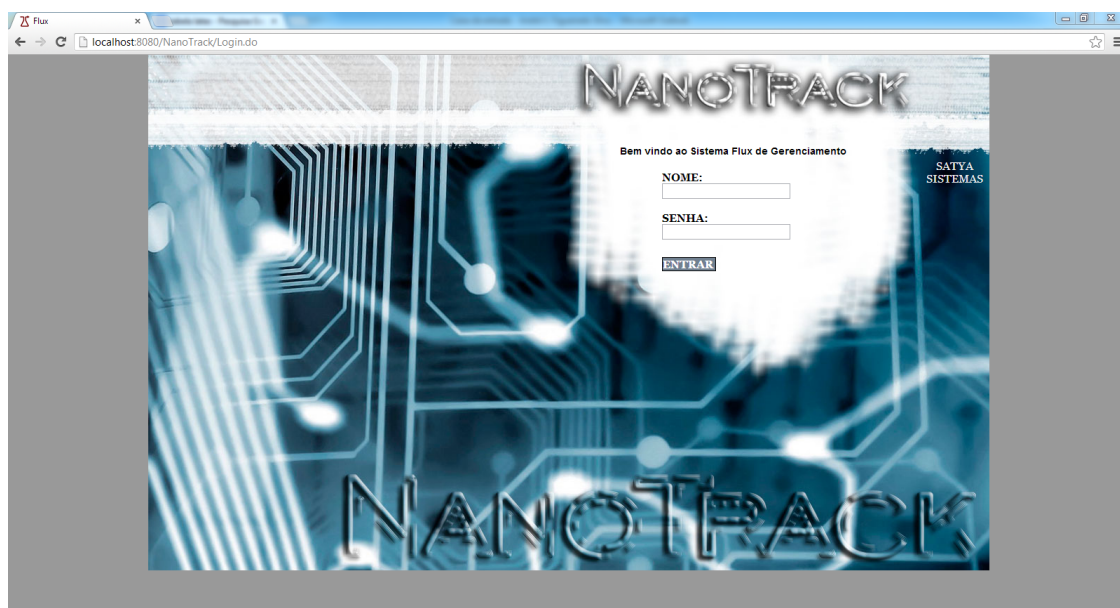
Neste trabalho foi desenvolvido o sistema *NanoTrack*, para o gerenciamento e análise de dados de experimentos de nanotecnologia utilizando o sistema Flux<sup>®</sup>, o qual utiliza uma abordagem baseada em *workflows* e um ambiente de simulação integrado à plataforma. Para a realização dos experimentos e validação do algoritmo de inferência que foi desenvolvido nesta dissertação foram utilizados a mesma amostra de dados do trabalho [Singulani et al., 2008]. Além destas amostras, também foram utilizados dados coletados junto ao Instituto Nacional de Ciência e Tecnologia em Nanodispositivos e

Semicondutores utilizados na síntese de nanodispositivos.

# Capítulo 4

## LIMS em nanotecnologia

Visando suprir a demanda de sistemas computacionais capazes de atender os laboratórios de nanotecnologia, neste trabalho foi desenvolvido o *NanoTrack*, vide Figura 4.1, um LIMS baseado na plataforma do sistema Flux® com um ambiente de simulação integrado. No *NanoTrack* foram implementados *workflows* de três laboratórios de nanotecnologia: *Pontos Quânticos*, *Laboratório de Semicondutores* e *Nanotubos de Carbono*. Eles foram criados através da aplicação *TWE - Together Workflow Editor* [Teamsolutions, 2011] e *Fluxar* [Thorun et al., 2013], ambos editores gráficos de apoio à construção do arquivo XML que representa o *workflow* e é interpretado pelo *NanoTrack*.



**Figura 4.1.** Interface de login da ferramenta NanoTrack.

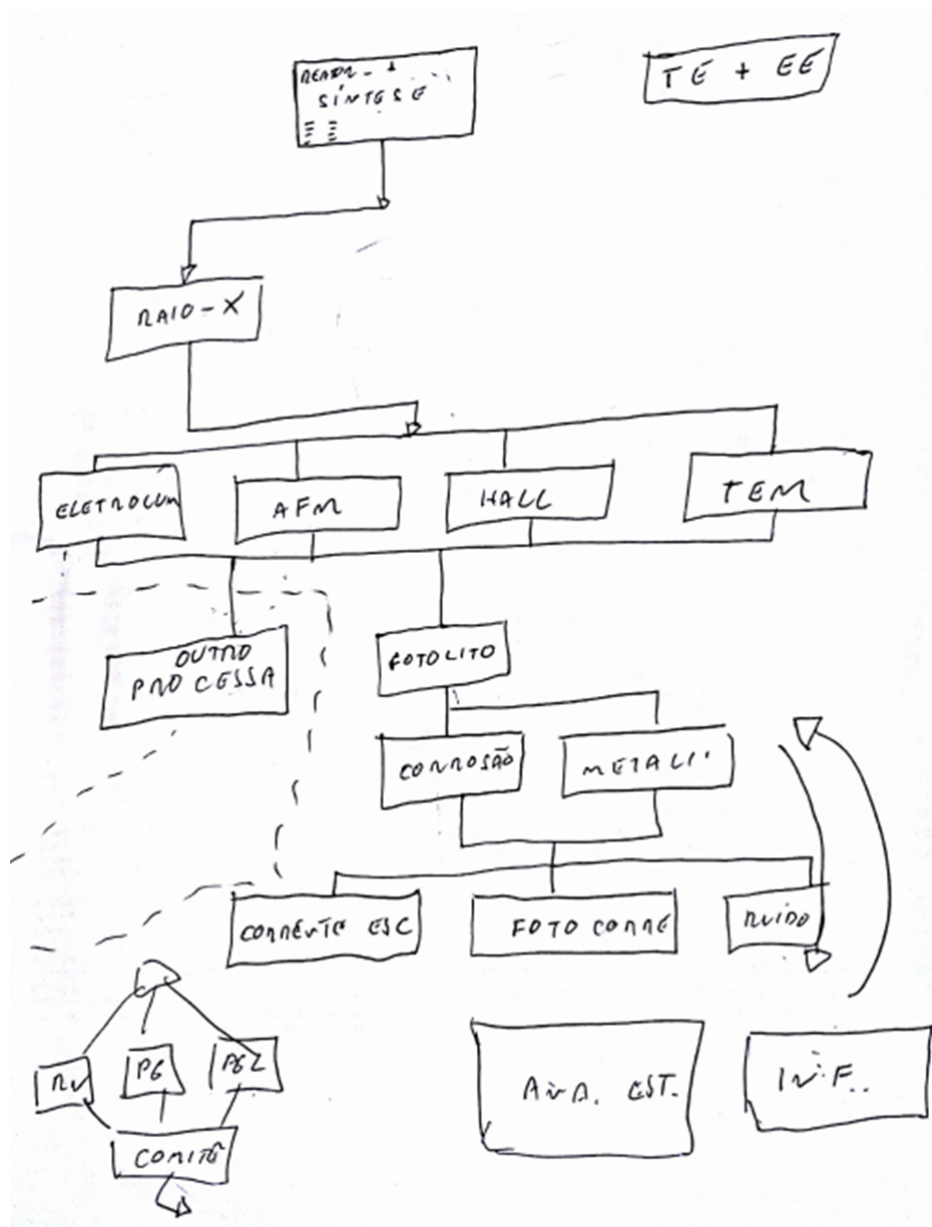
## 4.1 Modelagem

Cada laboratório possui características próprias de funcionamento e estas devem ser mapeadas na ferramenta. O primeiro passo para realizar a modelagem do laboratório é tomar conhecimento de como ele funciona, um passo-a-passo em todas suas etapas. Isto é feito com auxílio de algum membro do laboratório que tenha ciência de toda a estrutura e funcionamento do mesmo. Através da interação com os pesquisadores, o laboratório é modelado em forma de um fluxo de trabalho, *workflow*, onde as etapas identificadas dentro dos processos existentes nos laboratórios são representadas como atividades. Para cada dado acerca da atividade existe um atributo correspondente dentro do sistema. Nesta etapa, os materiais de apoio que os pesquisadores utilizam durante os experimentos também podem ser coletados junto ao membro do laboratório, visando automatizar todo o processo, a fim de centralizar as informações num mesmo ambiente computacional.

Veja na Figura 4.2 um esboço do funcionamento do Laboratório de Semicondutores (PUC-Rio) que foi modelado como um dos estudos de caso deste trabalho e será detalhado no próximo capítulo. Esta figura mostra um rascunho desenhado a mão, criado durante uma reunião com os pesquisadores do LabSem. O momento da interação entre os desenvolvedores da ferramenta e os membros do laboratório é dinâmico e os detalhes não podem passar despercebidos. É necessário que estes detalhes sejam anotados e o método mais fácil, neste caso, é através de rascunho com lápis e papel.

Em seguida, todo o material gerado e capturado durante as interações da etapa anterior deve ser codificado no formato XML para que o *NanoTrack* consiga interpretá-lo. Com este objetivo, dois editores visuais foram utilizados, *Together Workflow Editor* e o *Fluxar*, ambos com o mesmo propósito. O TWE é uma ferramenta completa para a criação de diagramas BPMN [Specification, 2006] (*Business Process Modeling Notation*, ou em português *Notação de Modelagem de Processos de Negócio*). Esta possui diversas funcionalidades e muitas destas não são aplicadas ao *workflow* interpretado pelo *NanoTrack*. Como consequência, a criação dos *workflows* fica sendo uma tarefa mais trabalhosa. Uma alternativa, que ainda está em desenvolvimento pelo laboratório LUAR, é a utilização da ferramenta *Fluxar* que já possui as principais funcionalidades implementadas e visa simplificar as etapas de criação encontradas no TWE.

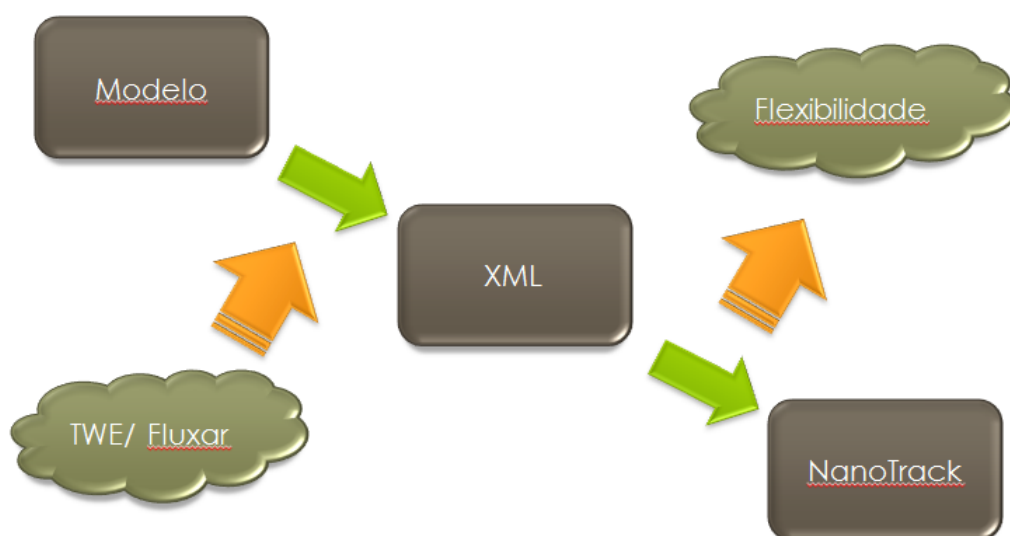
Durante a criação do *workflow* é possível, para cada atividade, determinar como os dados dos atributos das atividades serão armazenados no banco de dados. Por padrão, estes dados são armazenados na própria estrutura do sistema Flux®<sup>®</sup>, mas também permite que eles sejam armazenados em tabelas auxiliares especificadas pelo desenvolvedor. Para isto, o responsável por codificar o *workflow* em XML deverá



**Figura 4.2.** Esboço da modelagem do Laboratório de Semicondutores (PUC-Rio).

informar, durante a criação das atividades e atributos, que ele está utilizando tabela externa. Mesmo com as tabelas externas, o gerenciamento dos dados, bem como os relacionamentos, continua sendo feito pela plataforma Flux®. Somente a forma como o armazenamento é feito que é alterada.

Neste trabalho, em todas as atividades dos *workflows* gerados, a opção foi por armazenamento em tabelas externas. Esta escolha é justificada pelo fato da comunicação com os *plugins* ser facilitada, sendo necessário um menor número de consultas para



**Figura 4.3.** Processo de modelagem de um laboratório para a ferramenta NanoTrack.

obter os dados requisitados e utilizados por eles. Devido a esta escolha, foi necessário criar os *scripts* em SQL (*Structured Query Language*, ou em português *Linguagem de Consulta Estruturada*) para a criação das tabelas externas contendo todos os atributos pertencentes as atividades. Antes que o arquivo XPDL (XML Process Definition Language) seja inserido no *NanoTrack*, os *scripts* SQL gerados devem ser executados no banco de dados. Isto porque uma vez que o *workflow* esteja disponível no *NanoTrack* o usuário já terá condições de utilizá-lo e caso as tabelas do banco de dados não estejam disponíveis ocasionará erros no sistema.

Na Figura 4.3 é possível perceber uma visão geral de como é a etapa de Modelagem descrita nesta seção. Resumidamente, o modelo é gerado durante a interação com os pesquisadores do laboratório, através de um esboço de seu funcionamento. O *workflow* rascunhado deverá então ser digitalizado para o formato XPDL com o auxílio de ferramentas como o *Together Workflow Editor* e *Fluxar*. Em seguida, com o *workflow* já digitalizado, ele é incorporado ao *NanoTrack*, sendo que ele aceita incorporar diversos *workflows* e, por isto, é flexível.

## 4.2 Migração de dados

Uma das dificuldades dos laboratórios é quanto ao armazenamento dos dados gerados nos experimentos, muitos não tem um procedimento padrão definido e, por isto, os dados não chegam a ser registrados. Alguns laboratórios mantêm alguma forma de registro, mesmo que esta seja precária, tais como caderno de anotações ou métodos



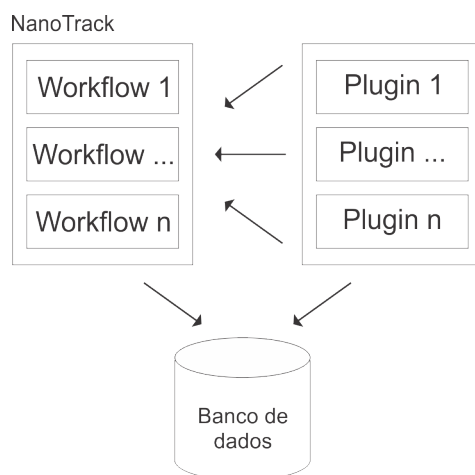
mais sofisticados, como planilhas eletrônicas. O importante é que se estas informações estão registradas de alguma maneira, elas também devem ser armazenadas no sistema implantado. Estes dados poderão ser cadastrados no *NanoTrack* através da própria aplicação, ou seja, através de um procedimento manual, ou caso existam muitos registros é aconselhável o desenvolvimento de algum *script* para realizar esta tarefa e popular as tabelas do banco de dados.

É importante que estes dados sejam cadastrados no sistema, pois além de ser um método padronizado e organizado irá permitir aos usuários da ferramenta possíveis buscas por experimentos realizados no passado. Obviamente, a tarefa de procurar por experimento é muito mais fácil quando se tem uma ferramenta computacional do que realizá-la como um processo manual.

### 4.3 Plugins

O *plugin* é um módulo de extensão capaz de adicionar funções a uma ferramenta hospedeira. Os *plugins* são acoplados ao sistema a fim de prover funcionalidades específicas que não eram atendidas pela ferramenta, tornando o desenvolvimento destes componentes independente do desenvolvimento da aplicação. Com isto, é possível que haja evoluções tanto por parte da ferramenta quanto por parte do *plugin* sem que um afete a estrutura do outro.

Na Figura 4.4 é mostrado os *plugins* como módulos independentes que interagem com o *NanoTrack*. Ela também mostra que eles podem compartilhar a mesma base de dados utilizada pelo *NanoTrack*, permitindo assim que os dados armazenados pelo



**Figura 4.4.** Estrutura genérica de interação entre o NanoTrack, plugins e banco de dados.

LIMS sejam utilizados pelos *plugins* para que eles possam realizar os processamentos necessários em suas operações.

### 4.3.1 Interação: Plugin x NanoTrack

A plataforma do *NanoTrack* já possui suporte para a chamada de processos que são executadas de modo independente. Para que um *plugin* seja introduzido na aplicação deve ser criada uma nova atividade no *workflow*, contendo as informações que serão passadas para o processamento do *plugin*. Uma vez que os atributos das atividades são preenchidos e o procedimento é confirmado pelo usuário, a aplicação cria um processo para a execução do *plugin*. Este processo é executado de modo síncrono, ou seja, a aplicação aguarda seu término para continuar o seu funcionamento.

O funcionamento do *plugin* não interfere diretamente no *NanoTrack*. Para a aplicação ele funciona como uma caixa preta, onde os dados de entrada são inseridos e é esperada uma saída. A saída obtida durante a execução do *plugin* é armazenada em um arquivo texto, que foi especificado como parâmetro pelo *NanoTrack*. Este arquivo texto será utilizado para exibir o resultados encontrados para o usuário da aplicação e sua saída pode ser texto sem formatação ou HTML (*HyperText Markup Language*, que significa *Linguagem de Marcação de Hipertexto*). A possibilidade de uso do HTML é bastante útil para que a saída possa ser formatada de uma maneira específica e que seja intuitiva para o usuário.

### 4.3.2 Consulta Externa

Um método de busca sugerido por pesquisadores dos laboratórios de nanotecnologia é uma busca com margem de erro. Este modelo de pesquisa proposta não irá apenas retornar os dados com os valores que atendem ao critério de busca como também irá retornar valores próximos que estejam dentro de uma margem de erro. A justificativa de utilizar este método de busca é devido à possibilidade de existência de experimentos com resultados semelhantes e que podem auxiliar o pesquisador.

Entretanto, por ser uma busca incomum, o sistema de busca padrão do *NanoTrack* não possui tal funcionalidade. Neste caso, pensando em prover uma nova funcionalidade específica para esta ferramenta, um *plugin* foi criado com este propósito. Ele funciona da seguinte forma:

Seja um conjunto de dados, como ilustrado na tabela 4.1, imagine que o pesquisador esteja interessado em amostras cuja *Altura Média* dos pontos quânticos seja **6,75**, mas as amostras que tenham uma variação de **0,10** para baixo ou para cima

também são aceitáveis. As consultas convencionais não permitem ao usuário ter esta flexibilidade, aceitando alguma margem de erro. Neste caso, o resultado da consulta convencional não traria nenhuma amostra para o pesquisador, sendo que na realidade existem amostras que atenderiam a necessidade dele.

<i>Amostra</i>	<i>Fluxos</i>			<i>Densidade</i>	<i>Altura Média</i>	<i>Desvio padrão</i>
	TM In	TM Al	TM Ga			
1	180	1	7,99	78	5,61	2,27
2	180	2	6,9	120	6,81	3,11
3	180	5	4,8	210	4,45	1,68
4	180	10	4	91	6,85	2,02
5	180	0	9,6	105	7,42	1,71
6	180	1	7,5	87	5,95	3,18
7	180	3,5	5,5	60	7,17	2,74
8	180	3,5	5,9	138	5,50	1,97
9	180	3,5	5,9	129	7,04	2,48
10	180	3,5	5,9	132	6,74	2,97

**Tabela 4.1.** Exemplo de base de dados

Com a utilização da consulta desenvolvida e encapsulada no plugin, o resultado da consulta para o exemplo descrito acima pode ser visto na tabela 4.2. Observe que a ordem em que os dados são exibidos depende da proximidade com o valor desejado, não importando se a variação é para baixo ou para cima.

<i>Amostra</i>	<i>Fluxos</i>			<i>Densidade</i>	<i>Altura Média</i>	<i>Desvio padrão</i>
	TM In	TM Al	TM Ga			
10	180	3,5	5,9	132	6,74	2,97
2	180	2	6,9	120	6,81	3,11
4	180	10	4	91	6,85	2,02

**Tabela 4.2.** Resultado obtido com utilização da consulta desenvolvida

Para atender esta carência, um *plugin*, capaz de executar consultas como a descrita acima, foi desenvolvido. Para que ele fosse integrado à ferramenta, foi criado no *workflow* uma atividade que requisitasse um processo externo. Esta atividade foi nomeada de *Consulta Externa* e toda vez que é executada ela interage com o *plugin* criado, passando os atributos da atividade como parâmetros para o *plugin*. Veja na imagem 4.5 como a atividade é representada dentro do *NanoTrack*.

Uma vez que o *plugin* recebe os parâmetros da ferramenta, ele irá realizar as operações necessárias, fazer consultas ao banco de dados a fim de extrair os dados solicitados pelo usuário, organizar os dados de modo a ordená-los para que os resultados

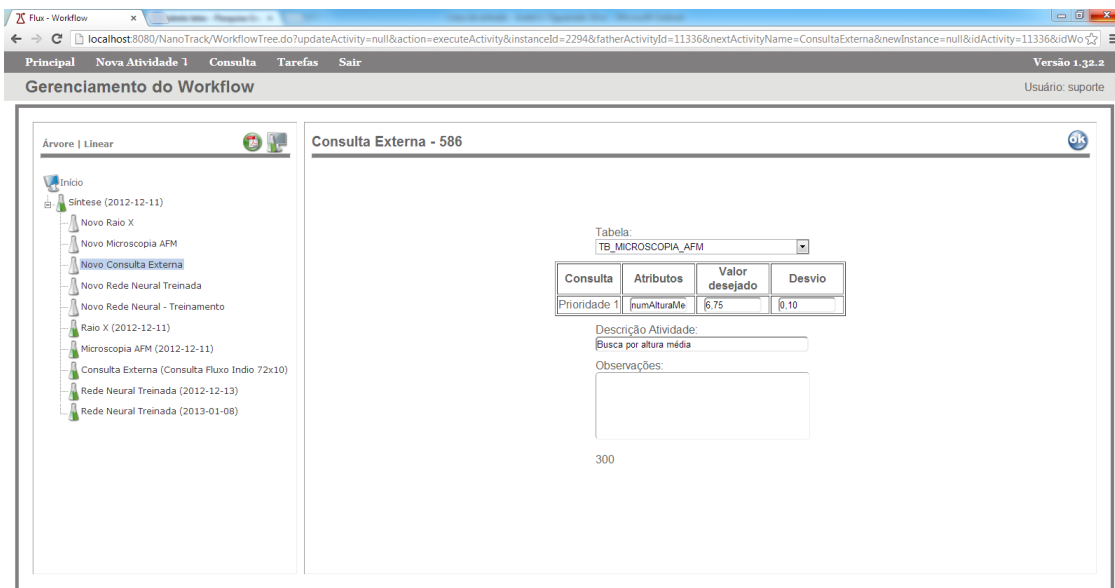


Figura 4.5. Atividade de consulta externa no NanoTrack.

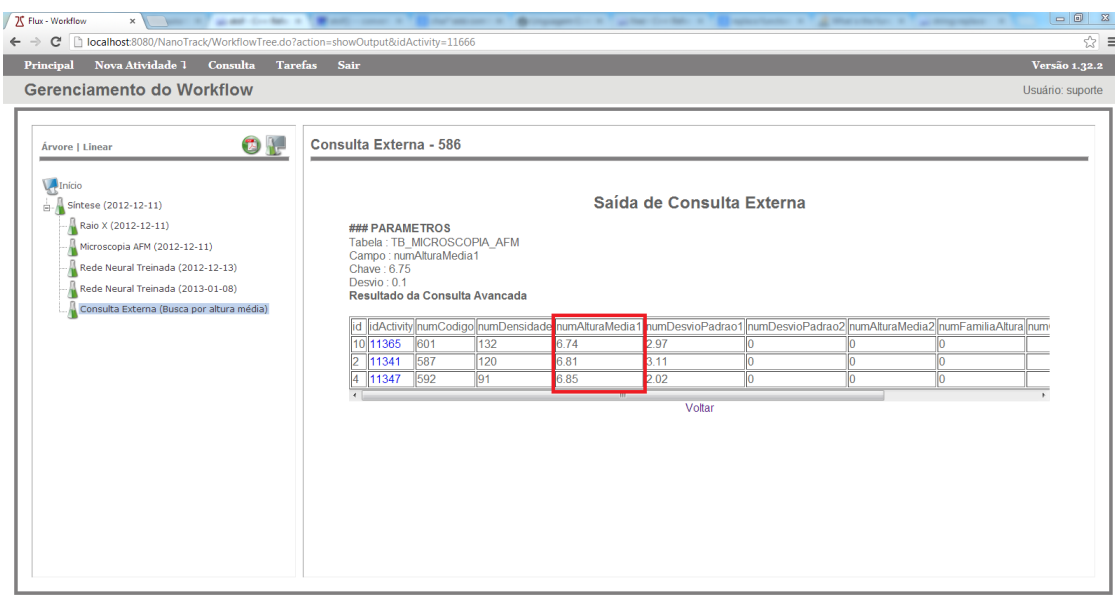


Figura 4.6. Resultado da saída do plugin de consulta externa.

que estejam mais próximos dos valores buscados tenham prioridade perante os demais registros e, por fim, gerar uma saída para que a ferramenta possa exibir os dados para o usuário. A imagem 4.6 mostra um exemplo de saída obtido pelo plugin após uma consulta.

Após a finalização do processo responsável pela consulta, o resultado é disponibilizado para o usuário para que ele possa visualizá-lo. Caso se interesse por algum

experimento retornado pela consulta, o usuário pode clicar no *link* disponível e visualizar todo o registro com todos os procedimentos realizados acerca do experimento selecionado. Isto lhe dá comodidade e maior facilidade para explorar experimentos que estejam no banco de dados e que podem servir de apoio a outros experimentos.

### 4.3.3 Plugin de Inferência (Rede Neural)

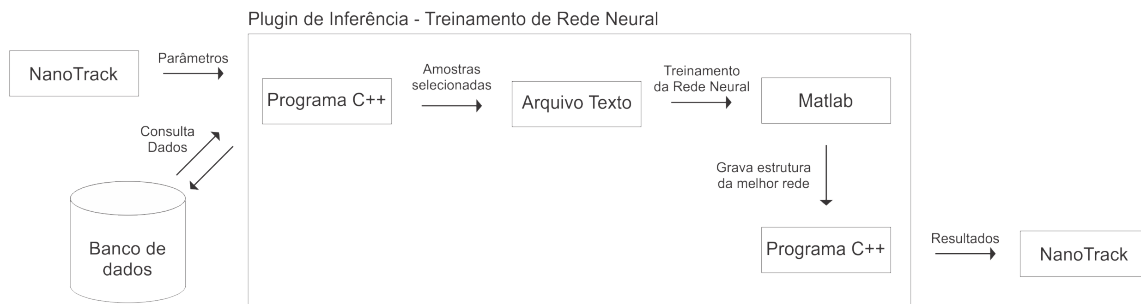
Como foi mostrado na seção 3.2, já existem diversos algoritmos de inferência desenvolvidos, cada um com uma especialidade diferente. É improvável que um único algoritmo de inferência seja capaz de resolver todos os problemas de simulação existentes nos laboratórios de nanotecnologia, por isto, o ambiente de simulação desenvolvido neste trabalho propõe uma abordagem baseada em *plugins*. Para acoplar qualquer simulador à ferramenta, basta que ele seja encapsulado em forma de um *plugin* e uma atividade seja criada no *workflow*.

Os simuladores desenvolvidos neste trabalho foram baseados em redes neurais artificiais e implementados em *Matlab*. *Matlab* é uma linguagem para computação científica de alto desempenho, desenvolvida para realizar processamentos de cálculos numéricos, cujo o elemento básico é uma matriz [Guide, 1998]. *Matlab* possui um ambiente composto por uma série de funções pré-definidas para cálculo, leitura e escrita de arquivos e visualização.

Nesta biblioteca de funções, já existe um algoritmo genérico de rede neural disponível. Para utilizá-lo é necessário apenas customizá-lo, ajustando as configurações da rede neural, tais como: número de neurônios, número de camadas, função de ativação, além de determinar quais são os parâmetros de entrada e quais os de saída. Uma vez que a rede neural é configurada, basta passar uma matriz contendo os dados para treinamento, validação e teste. Com base nos dados passados para a rede neural, o algoritmo irá ajustar os pesos da rede neural a fim de encontrar a melhor rede com a maior precisão.

Obter uma configuração muito boa para a rede neural, onde a taxa de erro seja bastante baixa, não é trivial. A melhor configuração de um experimento com rede neural, normalmente, não será a mesma para outra. Assim, para tentar encontrar uma boa configuração é necessário executar o treinamento diversas vezes e, em cada uma delas, variar os parâmetros de sua estrutura, como por exemplo o número de neurônios em uma determinada camada e o número de camadas da rede neural.

Nos simuladores desenvolvidos, para encontrar as melhores configurações das rede neurais foi implementado um algoritmo que realiza este procedimento de modo dinâmico. Com isto, para um mesmo conjunto de dados selecionado, a rede neural é execu-



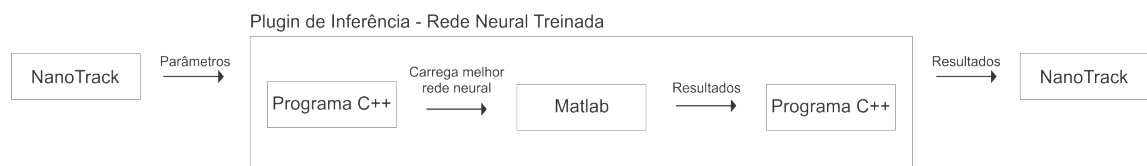
**Figura 4.7.** Protótipo da interação entre NanoTrack e Plugin de treinamento da rede neural.

tada diversas vezes para várias configurações diferentes. Sempre o resultado da melhor rede neural é armazenado para que ao final do treinamento a melhor rede neural seja gravada em um arquivo e, assim, ela poderá ser utilizada posteriormente.

A rede neural é o simulador que irá auxiliar o pesquisador, entretanto ele pode precisar ser calibrado na medida que novos experimentos são realizados, a fim de atualizar o aprendizado da rede neural com os novos resultados. Para permitir a utilização do simulador e também que ele seja calibrado, foram desenvolvidos dois tipos de *plugins*: “rede neural treinamento” e “rede neural treinada”. O primeiro é a etapa de aprendizagem do simulador. Esta não gera resultados de simulação, apenas atualiza a calibração da rede neural utilizando desde os dados mais antigos até os mais recentes. O segundo é o simulador propriamente dito e este gera resultados. Uma vez que o usuário especifica os parâmetros, o simulador irá processar os dados e retornar o resultado da simulação.

A fim de encapsular o algoritmo da rede neural e gerar um arquivo binário para ser incorporado ao *NanoTrack*, os *plugins* desenvolvidos foram implementados na linguagem C++. Veja na Figura 4.7 um protótipo da interação entre a ferramenta *NanoTrack* e o *plugin* desenvolvido para a calibração da rede neural. Os *plugins* intermediam a comunicação entre o *NanoTrack* e o *Matlab*. Nesta figura, os parâmetros são passados do *NanoTrack* para o *plugin*, que em seguida realiza uma busca pelos dados de experimentos cadastrados através da ferramenta *NanoTrack*, estes dados são carregados no *plugin* e em seguida são repassados para o *Matlab*, que os utilizam para realizar o treinamento da rede neural. Quando o processo de treinamento é concluído, a rede neural fica disponível para ser utilizada.

Na Figura 4.8 é mostrado o protótipo da interação entre a ferramenta *NanoTrack* e o *plugin* de simulação. Este *plugin* possui o funcionamento mais simples do que o anterior, pois ele irá utilizar a rede neural que já foi treinada anteriormente. Através do *NanoTrack*, o usuário irá informar os parâmetros de entrada e o simulador irá processar estes dados e retornar a saída. Em seguida, os resultados obtidos ficam disponíveis para



**Figura 4.8.** Protótipo da interação entre NanoTrack e Plugin para utilização de uma rede neural já treinada.

o usuário na ferramenta *NanoTrack*.





# Capítulo 5

## Estudos de casos

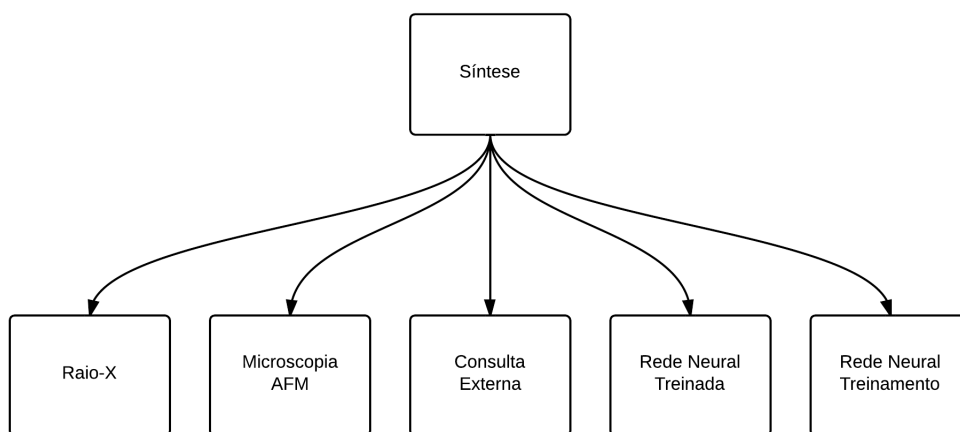
Neste trabalho foram desenvolvidos três estudos de casos de laboratórios de nanotecnologia: **Pontos Quânticos**, **LabSem - Laboratório de Semicondutores (PUC-Rio)** e **Nanotubos de Carbono (Física-UFMG)**. O primeiro laboratório mapeado para o desenvolvimento deste projeto foi modelado a partir de conhecimentos e dados de experimentação em Pontos Quânticos, com o intuito de aprender sobre a plataforma Flux® e também serviu como protótipo para aplicação dos primeiros *plugins* desenvolvidos. O segundo e terceiro caso de estudos são mapeamentos com dados reais de todo o processo de experimentação realizado nos respectivos laboratórios, sendo que o LabSem já possuía dados armazenados, possibilitando o desenvolvimento de um ambiente de simulação inovador e com uma precisão muito alta.

### 5.1 Pontos Quânticos

Com o objetivo de construir o *NanoTrack* e familiarizar com a plataforma Flux® e o software de apoio *Together Workflow Editor*, a modelagem do processo de experimentação de Pontos Quânticos foi realizada. As etapas deste processo foram identificadas e organizadas de forma ordenada. Além disto, este *workflow* foi utilizado como protótipo para implantar os *plugins* desenvolvidos tanto para a *Consulta Externa* quanto para os simuladores desenvolvidos baseados em redes neurais artificiais.

#### 5.1.1 Modelagem

A modelagem deste *workflow* foi baseada na experiência anterior da utilização de dados de Pontos Quânticos. Através da experiência nos procedimentos realizados nos laboratórios de síntese e caracterização de pontos quânticos foi possível organizar e estruturar



**Figura 5.1.** Workflow de experimentação de Pontos Quânticos.

em forma de fluxo de trabalho, vide Figura 5.1, a estrutura final do *workflow*.

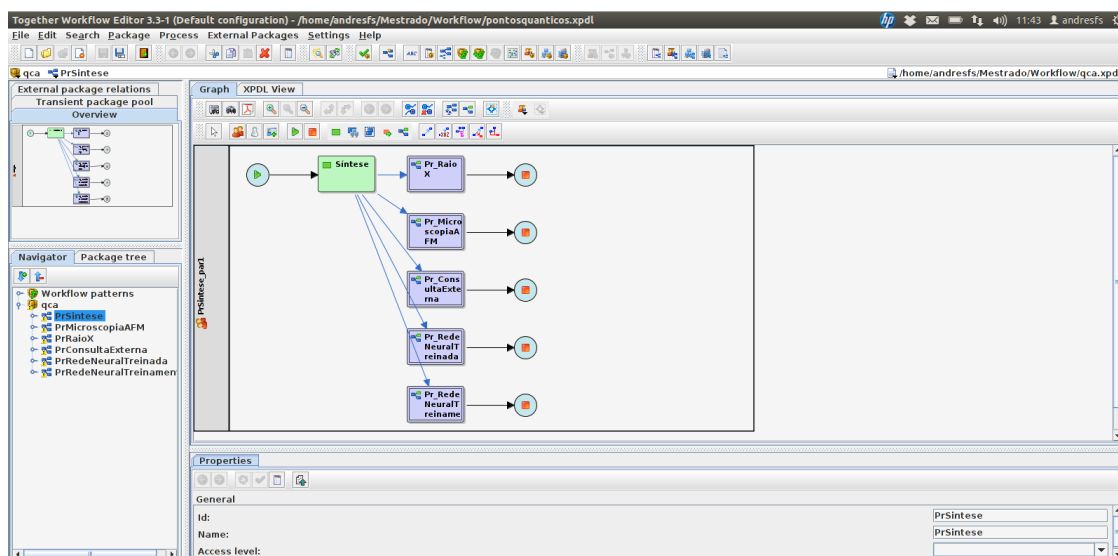
Os processos identificados nestas etapas podem ser classificados em dois grupos: síntese e caracterização. Além destes, um terceiro grupo de atividades foi inserido formando a parte dos módulos incorporados à ferramenta, os *plugins*. Desta forma, as seguintes atividades identificadas e seus respectivos grupos estão apresentadas na tabela 5.1.

<i>Grupo</i>	<i>Atividade</i>
Síntese	Síntese
Caracterização	Raio-X
	Microscopia AFM
Plugin	Consulta Externa
	Rede neural treinada
	Rede neural treinamento

**Tabela 5.1.** Lista de atividades identificadas durante a criação do workflow de Pontos Quânticos.

Para cada processo identificado foi criada uma atividade que o representasse. Estas atividades são compostas por propriedades que representam os dados gerados durante a sua execução e estas deverão ser armazenados no banco de dados da ferramenta a fim de possibilitar futuras consultas e auxiliar os algoritmos de inferência na descoberta de conhecimento. Estas propriedades foram obtidas por meio do trabalho [Singulani et al., 2008], através de uma planilha eletrônica contendo um cadastro de diversos experimentos realizados pelo *LabSem* da PUC-Rio.

Após modelar e identificar as propriedades do processo de experimentação de Pontos Quânticos é preciso codificá-lo para o formato XPDL que será interpretado pelo *NanoTrack*. Para isto a ferramenta *Together Workflow Editor (TWE)* foi utilizada, veja



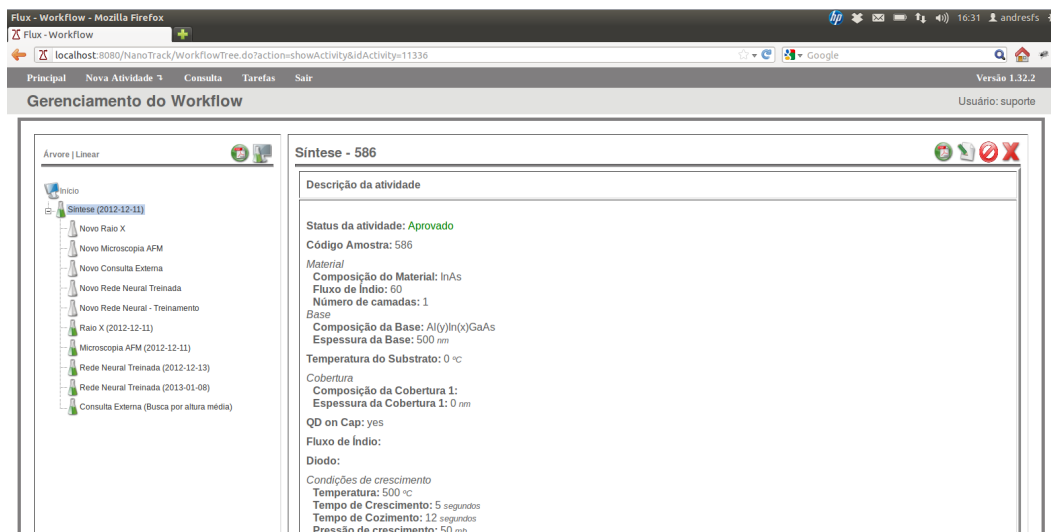
**Figura 5.2.** Interface do Together Workflow Editor com o workflow de Pontos Quânticos sendo codificado para XPD.

Id	Name	Data type	Initial v...
dscComposicaoBase	Composição da Base	Basic type -> String	
dscComposicaoCob...	Composição da Cobertura 1	Basic type -> String	
dscComposicaoCob...	Composição da Cobertura 2	Basic type -> String	
dscComposicaoMat...	Composição do Material	Basic type -> String	
numCodigo	Código Amostra	Basic type -> Integer	
dscDiodo	Diodo	Basic type -> String	
numEspessuraBase	Espessura da Base	Basic type -> Float	
numEspessuraCobe...	Espessura da Cobertura 1	Basic type -> Float	
numEspessuraCobe...	Espessura da Cobertura 2	Basic type -> Float	
dscFluxoIndio	Fluxo de Índio	Basic type -> String	
numFluxoIndio	Fluxo de Índio	Basic type -> Float	
imgSíntese	Imagem de Síntese	Declared type -> file	
numML	ML	Basic type -> Float	
numNivelDopagem	Nível de dopagem	Basic type -> Float	
numNumeroCamadas	Número de camadas	Basic type -> Integer	
dscPL	PL	Basic type -> String	
numPressaoCresci...	Pressão de crescimento	Basic type -> Float	
indQDonCap	QD on Cap	Declared type -> checkbox	
indLT	Sim	Declared type -> checkbox	
indRT	Sim	Declared type -> checkbox	
dscSustr	Sustr	Basic type -> String	
numFluxoTMAI	TM AI	Basic type -> Float	
numFluxoTMGa	TM Ga	Basic type -> Float	
numFluxoTMIn	TM In	Basic type -> Float	
numTemperaturaCr...	Temperatura	Basic type -> Float	
numTemperaturaSu...	Temperatura do Substrato	Basic type -> Float	
numTempoCozimento	Tempo de Cozimento	Basic type -> Float	
numTempoCrescime...	Tempo de Crescimento	Basic type -> Float	
dscTipoDopagem	Tipo de dopagem	Basic type -> String	
indTreinamento	Treinamento	Declared type -> checkbox	

**Figura 5.3.** Trinta atributos que compõem a atividade de síntese do workflow de Pontos Quânticos.

na Figura 5.2 a interface TWE após a codificação deste *workflow*. Durante a criação das atividades no TWE, as propriedades identificadas anteriormente são codificadas em forma de atributos. Veja na Figura 5.3 os atributos que compõem a atividade de Síntese.

A Figura 5.4 exibe a interface da atividade de síntese carregada através do *NanoTrack* com os dados de uma amostra sendo exibidos no lado direito da imagem. Além



**Figura 5.4.** Interface do Nanotrack exibindo os dados cadastrados em uma atividade de síntese.

destes dados, é possível visualizar do lado esquerdo a estrutura do *workflow* desenvolvido e as demais atividades prontas para serem acessadas ou criadas.

#### 5.1.1.1 Tabelas Externas

Os atributos das atividades de síntese e caracterização foram marcados para utilizarem tabelas auxiliares durante a codificação para XPD. Esta escolha visa facilitar e simplificar o acesso dos *plugins* aos dados armazenados pelo *NanoTrack*. Para cada uma destas atividades foi criada uma tabela responsável por armazenar os seus respectivos dados. Os nome das tabelas criadas e sua respectiva atividade pode ser visualizada na tabela 5.2.

<i>Atividade</i>	<i>Tabela</i>
Síntese	TB_SINTESE
Raio-X	TB_RAIO_X
Microscopia AFM	TB_MICROSCOPIA_AFM

**Tabela 5.2.** Tabelas externas criadas para as atividades do workflow de Pontos Quânticos

#### 5.1.2 Importação de dados

Todos os registros de experimentos disponível na planilha eletrônica foram importados para o banco de dados do *NanoTrack*. Como havia muitos experimentos nesta planilha, esta tarefa foi automatizada por meio de um *script* desenvolvido. Além de ser mais

rápido a inserção dos registros no banco de dados, a utilização de um procedimento automático evita a inserção de erros que poderiam ser gerados durante o procedimento manual.

Para que o *NanoTrack* reconheça os dados que foram importados de maneira automatizada, o *script*, além de preencher as tabelas auxiliares criadas para as atividades, precisa também manipular as tabelas da plataforma Flux®. Isto é necessário para que os registros de controle interno sejam gerados adequadamente e o *NanoTrack* consiga manipulá-los normalmente em um momento futuro.

### 5.1.3 Consulta Externa

O *plugin* para realizar consultas com margem de erro apresentado na seção 4.3.2 foi implantado neste *workflow*. A atividade que o representa é chamada de “Consulta Externa” e realiza busca por experimentos nas atividades de síntese e caracterização modeladas neste *workflow*.

### 5.1.4 Ambiente de simulação

Neste *workflow* o ambiente de simulação é composto por dois *plugins*, um para *treinamento do simulador* e outro que é o *simulador*, ambos implementados em *Matlab*.

Através da utilização do simulador, o usuário da ferramenta poderá ajustar os parâmetros de entrada à sua maneira e o simulador irá processar estes dados e retornar o resultado. Neste simulador foram implementados três redes neurais capazes de inferir informações sobre a experimentação em *Pontos Quânticos*, tais como: densidade, altura média e desvio padrão.

#### 5.1.4.1 Treinamento da Rede Neural Artificial

A rede neural desenvolvida para simular os resultados de *Pontos Quânticos* foi criada a partir da biblioteca de funções disponível no *Matlab*, como explicado na seção 4.3.3. O primeiro passo para treiná-la foi identificar os parâmetros que iriam ser utilizados como entrada e saída. Conforme apresentado na Figura 2.3 da seção 2.4, a rede neural recebe os dados de entrada e, durante o seu processo de aprendizado, compara a saída obtida com os valores de saída esperada. Com base no erro obtido, que é calculado através da diferença das saídas, a rede neural irá ajustar os pesos que conectam os neurônios artificiais a fim de minimizar este erro.

Através do auxílio de pesquisadores que trabalham com o crescimento de *Pontos Quânticos*, foram identificados os parâmetros determinantes para o processo de cresci-

mento e que seriam utilizados como entrada da rede neural, além dos parâmetros de saída. Neste caso, foram utilizados alguns atributos que o pesquisador entende que irá influenciar diretamente na saída. Na tabela 5.3 está descrito a lista dos atributos que foram utilizados para treinar a rede neural e sua classificação que indica se o atributo é de entrada ou saída.

Os resultados de uma única rede neural para inferir os dados das três saídas, mostradas na tabela 5.3, não foram bons. Por isto, foram implementadas três redes neurais. Cada rede neural é responsável por inferir uma saída diferente. Entretanto, o algoritmo de treinamento da rede neural é o mesmo para todas elas.

<i>Atributo</i>	<i>Atividade</i>	<i>Classificação</i>
Fluxo de Índio (sccm)	Síntese	Entrada
Espessura da Base (nm)		
Temperatura de Crescimento (°C)		
Tempo de Crescimento (s)		
% Índio	Raio-X	
% Alumínio		
Densidade ( $10^8 QD/cm^2$ )	Microscopia AFM	Saída
Altura Média (nm)		
Desvio padrão (nm)		

**Tabela 5.3.** Lista de atributos de entrada e saída utilizados na rede neural.

Depois de identificar os atributos que são utilizados como entradas e saídas, é preciso gerar um arquivo com os dados dos experimentos que possua apenas estes atributos. Este arquivo gerado será acessado pelo *Matlab* e as colunas existentes no arquivo serão identificadas como entradas ou saídas. Em seguida, é realizado uma normalização dos dados. Esta normalização tem o intuito de reduzir a discrepância entre os valores dos atributos, sem prejudicar os dados.

Normalmente, os dados coletados são separados em duas categorias:

- Conjunto de treinamento: serão utilizados para o treinamento da rede neural;
- Conjunto de teste: serão utilizados para verificar o desempenho sob condições reais de utilização.

Além desta divisão, foi utilizado uma subdivisão do conjunto de treinamento, criando um conjunto de validação. Este novo conjunto é utilizado para verificar a eficiência da rede neural quanto a sua capacidade de generalização durante o processo de treinamento. Ele também pode ser utilizado como critério de parada do treinamento.

Cada conjunto de dados possui amostras diferentes. Do total das amostras carregadas no arquivo, 85% pertencem ao conjunto de treinamento, sendo que 15% deles é do subconjunto de validação. Os 15% restantes são referentes ao conjunto de teste. Esta distribuição de amostras nos conjuntos são parâmetros ajustáveis no algoritmo e com estes valores foi possível obter resultados satisfatórios, mas que podem ser modificados durante novas calibrações da rede neural.

Em seguida, é realizado o treinamento da rede. A etapa foi dividida em três partes. A primeira irá configurar a estrutura da rede, com o número de camadas e o número de neurônios em cada camada. A segunda é responsável por realizar o treinamento propriamente dito, etapa de aprendizado. E a última irá verificar se o resultado da rede neural atual é melhor que as demais redes neurais treinadas anteriormente. Caso seja, ela será armazenada. Este procedimento é realizado diversas vezes, variando o número de camadas entre uma e duas. Sendo que na primeira camada pode-se ter entre 2 a 15 neurônios e na segunda camada de 0 a 5 neurônios.

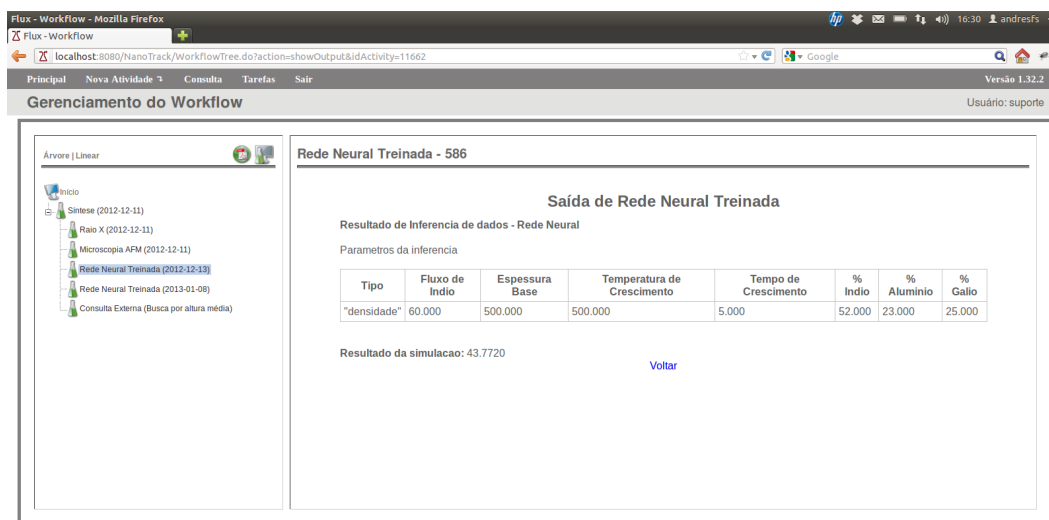
Após a conclusão da etapa de treinamento, a melhor rede neural obtida estará armazenada e em seguida os resultados obtidos por ela são gravados em arquivos para poderem ser utilizadas posteriormente.

#### 5.1.4.2 Utilizando a Rede Neural Artificial

A utilização da rede neural depende do seu prévio treinamento, uma vez que ela será utilizada para inferir os dados. Nesta etapa, a melhor rede neural obtida no passo anterior, que será utilizada como simulador, é carregada na memória do *Matlab*. Com isto, todo o ambiente que foi gerado durante a criação da rede neural fica disponível. Desta forma, a rede neural estará à disposição para ser utilizada.

Para utilizar a rede neural o usuário deverá informar os valores de entrada (os mesmos parâmetros de entrada da tabela 5.3) que ele deseja simular. Como os dados utilizados durante o treinamento foram normalizados, é importante que os dados a serem simulados também sejam normalizados e, para que o mesmo padrão de normalização seja seguido durante a simulação, é importante utilizar os mesmos dados do treinamento que estão disponíveis no ambiente carregado do arquivo. Em seguida, os dados de entrada do simulador informado pelo usuário é agrupado aos dados de treinamento e é realizada a normalização destes.

Após a normalização dos dados, o simulador da rede neural é invocado e os valores de entrada, já normalizados, são passados por parâmetro. A simulação é realizada de maneira extremamente rápida e o resultado obtido pela rede neural fica disponível, bastando apenas realizar o processo inverso da normalização para que os valores obtidos



**Figura 5.5.** Utilização da rede neural treinada dentro do ambiente de simulação do Nanotrack.

sejam condizentes com os valores reais.

Veja na Figura 5.5 um exemplo de simulação realizado através do *NanoTrack*. Na figura, os atributos utilizados como entrada na rede neural estão listados na tabela e, após a tabela o resultado obtido para estes parâmetros pode ser visto no campo “Resultado da simulacao”.

#### 5.1.4.3 Resultados

Como discutido anteriormente, não foi possível treinar uma única rede neural que obtivesse resultados satisfatórios para todas as saídas desejadas. Por isto, foi necessário treinar uma rede neural para cada saída. Com isto, realizamos o treinamento de três redes neurais que simulam: **densidade**, **altura média** e **desvio padrão** do processo de crescimento de pontos quânticos. Veja na tabela 5.4 como ficou a composição de cada camada escondida (número de neurônios) das redes neurais após o treinamento.

<i>Rede Neural</i>	<i>Primeira camada</i>	<i>Segunda camada</i>
Altura média	13	3
Densidade	6	4
Desvio padrão	14	5

**Tabela 5.4.** Quantidade de neurônios em cada camada da Rede Neural

A seguir serão exibidos os resultados obtidos contendo três gráficos em cada um deles. Os gráficos referem-se às etapas de treinamento, validação e teste. Sendo que



a linha de cor vermelha (tracejada - ponto) representa o valor obtido pela rede neural durante a simulação e a de cor azul (contínua - bolinha) é o valor esperado.

A rede neural de inferência de **densidade** obteve resultados com altíssima precisão, sendo que o erro percentual absoluto médio (*MAPE* - *Mean Absolute Percentage Error*) durante a etapa de teste foi de aproximadamente 2% e o coeficiente de correlação é bastante próximo de 1, o que indica que o simulador é capaz de gerar resultados muito próximo do mundo real. Veja na Figura 5.6 os gráficos obtidos durante as etapas.

O *MAPE* é uma medida de precisão utilizada pela estatística para estimativa de tendências. Ele representa a média de todos os erros absolutos percentuais, fornecendo uma indicação do tamanho médio do erro, expresso como uma porcentagem do valor observado. Veja a fórmula do MAPE abaixo, onde  $t_i$  representa o valor real e  $o_i$  o valor obtido com a simulação.

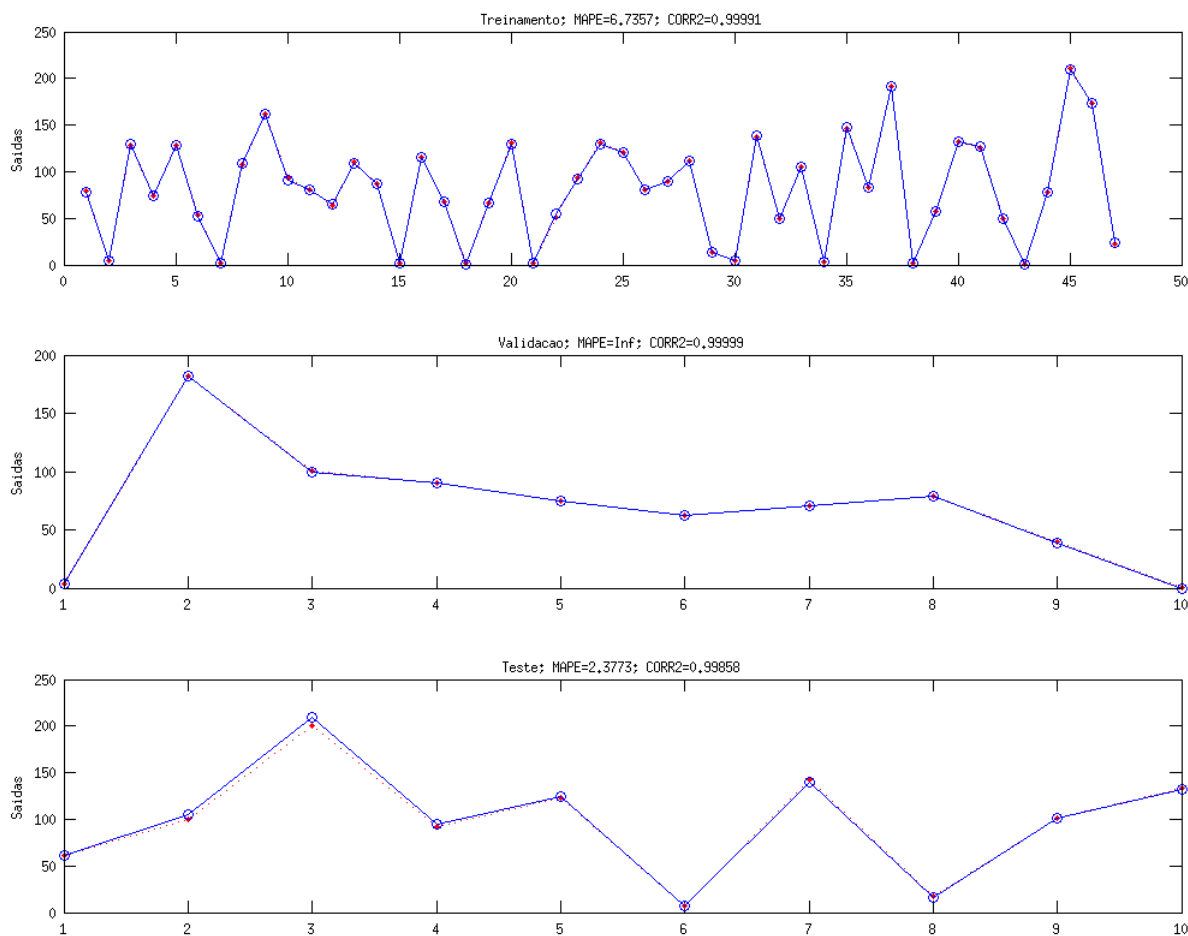
$$MAPE = 100 \times \frac{1}{n} \sum_{i=1}^n \left| \frac{t_i - o_i}{t_i} \right|$$

Veja na Figura 5.7 os resultados da rede neural desenvolvida para a inferência de **altura média**. Os resultados foram satisfatórios com um erro de aproximadamente 11% durante a etapa de teste. A correlação obteve um valor de 0.97.

E por último é mostrado na Figura 5.8 os resultados da rede neural desenvolvido para a inferência de **desvio padrão**. Os resultados desta rede não são satisfatórios, apresentando um erro de quase 30% e uma correlação de 0,15.

A obtenção de bons resultados pela rede neural depende muito da qualidade dos dados que são utilizados na etapa de treinamento. Por isto, é necessário que antes de treinar a rede neural seja feito um tratamento nos dados que serão utilizados. Este tratamento é basicamente um filtro onde as amostras boas são selecionadas para o treinamento e as amostras ruins são desclassificadas. Normalmente o pesquisador consegue distinguir uma amostra boa e uma amostra ruim pelos resultados obtidos. Durante a etapa de experimentação podem ocorrer eventos que prejudicam a qualidade do experimento e, conseqüentemente, o resultado. Quando ocorrem tais eventualidades o pesquisador deve marcar o experimento como uma amostra cujo resultado é ruidoso.

Com as amostras ruins marcadas fica mais fácil obter boas redes neurais, capazes de realizar a simulação com boa precisão. Se observarmos os resultados do treinamento da rede neural para densidade e compará-lo com os resultados da rede neural para desvio padrão existe um contraste muito grande com a qualidade dos resultados. Se houvesse um tratamento de dados de um especialista é provável que seria possível obter redes neurais melhores tanto para a saída de altura média quanto para a de desvio padrão.



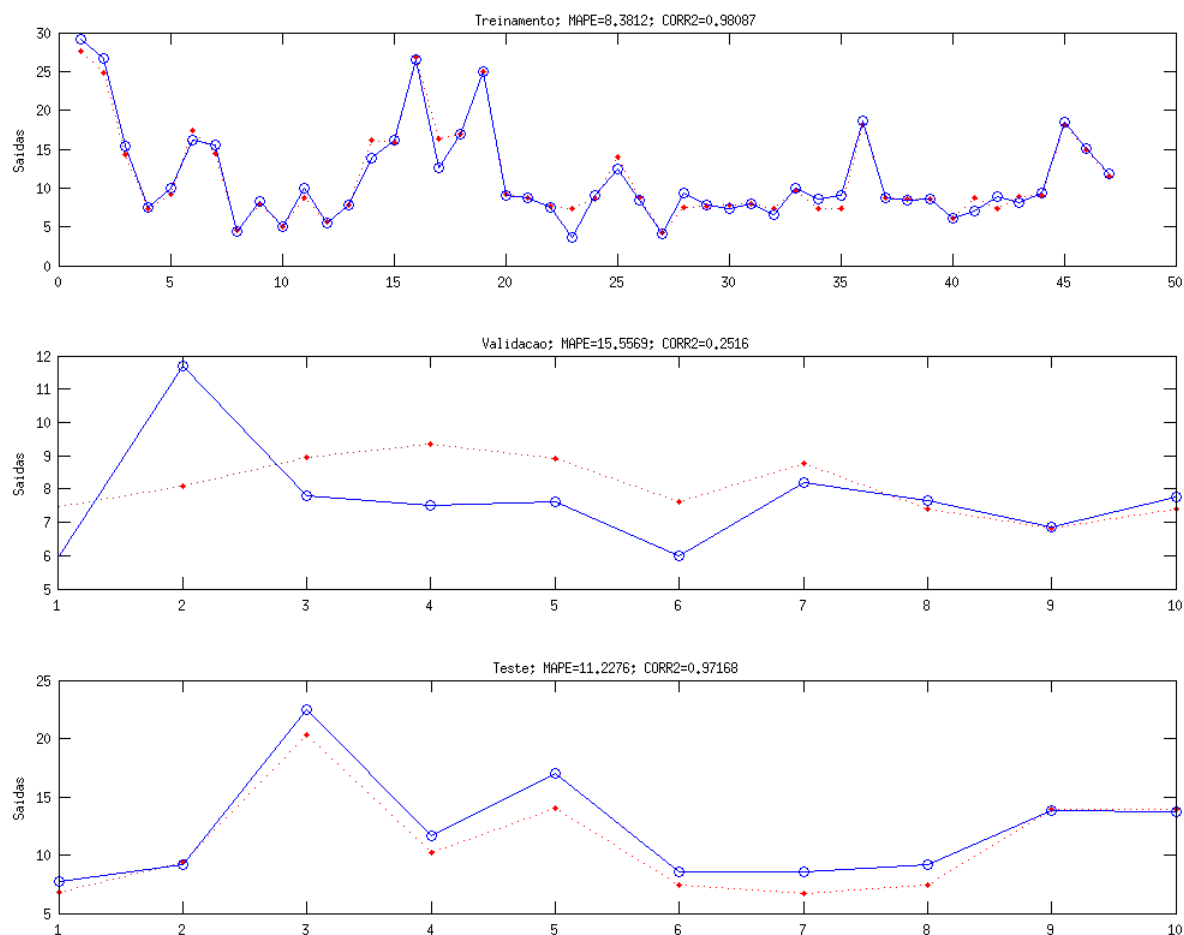
**Figura 5.6.** Resultado da rede neural nas etapas de treinamento, validação e teste para a saída de Densidade.

Como o objetivo deste *workflow* é mostrar a capacidade de integração de um ambiente de simulação com o LIMS o foco não foi aprimorar a rede neural para obter os melhores resultados. Outro fator determinante é que este *workflow* é um protótipo utilizado para os primeiros testes com o *NanoTrack*.

#### 5.1.4.4 Integração do plugin no workflow

Para cada um dos dois *plugins* desenvolvidos foi criada uma atividade contendo os atributos necessários para o seu funcionamento.

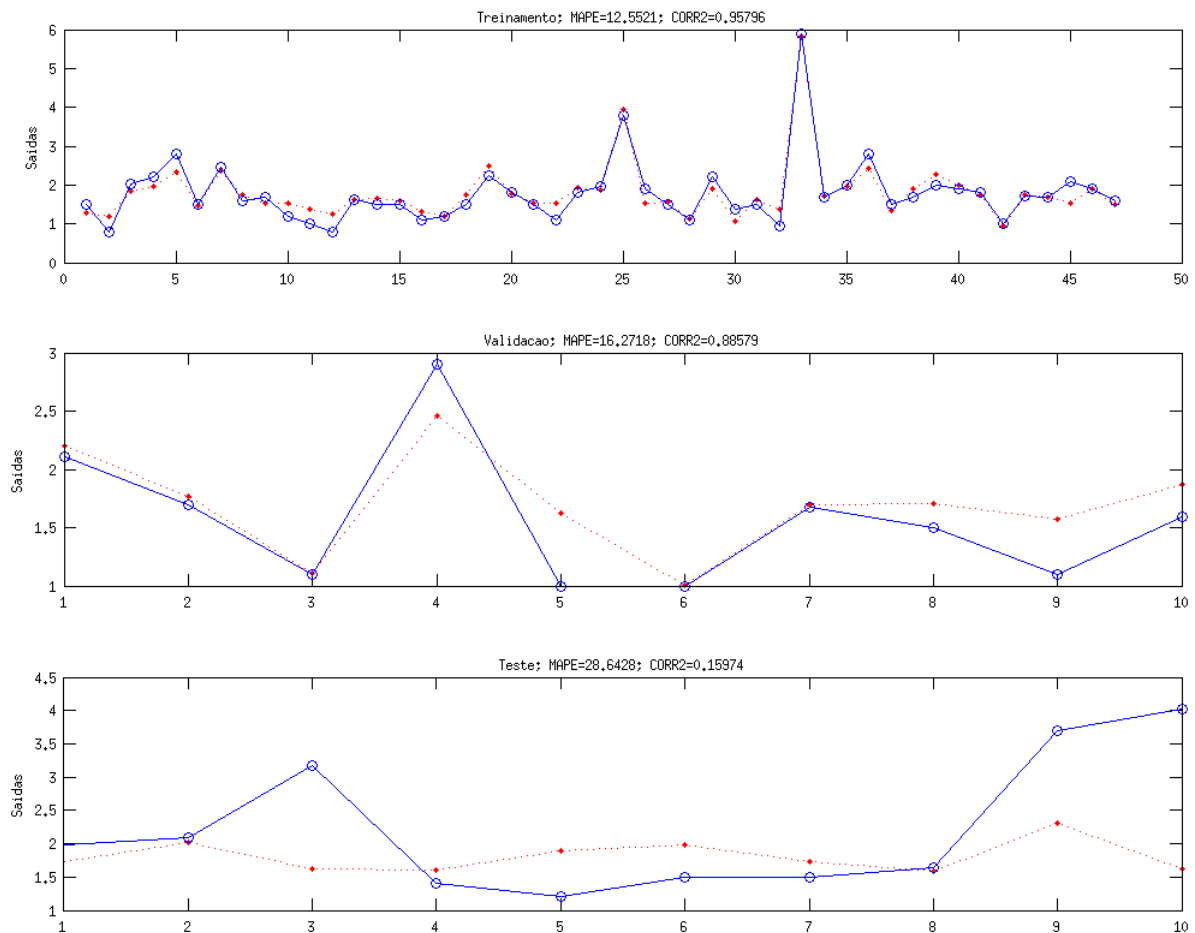
A atividade de treinamento do simulador é chamada no *workflow* de “Rede Neural - Treinamento” e ela é composta por dois atributos: *Tipo de Saída* e *Observações*. O *Tipo de Saída* irá determinar qual rede neural deverá ser treinada, sendo que as opções deste atributo são: densidade, altura média e desvio padrão. Uma vez que estes dados são informados, a atividade irá invocar o *plugin* para que ele possa realizar o



**Figura 5.7.** Resultado da rede neural nas etapas de treinamento, validação e teste para a saída de Altura Média.

treinamento da rede neural.

Já a atividade do simulador é denominada no *workflow* como “Rede Neural Treinada” e possui nove atributos: *Tipo de Saída*, *Fluxo de Índio*, *Espessura da Base*, *Temperatura de Crescimento*, *Tempo de Crescimento*, *% Índio*, *% Alumínio*, *% Gálio* e *Observações*. O *Tipo de Saída* possui a finalidade de determinar qual rede neural utilizar para a simulação. Os demais atributos são os parâmetros utilizados como entrada na rede neural, conforme mostrado na tabela 5.3, exceto os atributos *% Gálio* e *Observações* que foram inseridos com o intuito de manter um cadastro completo da simulação. Após informar os dados, a atividade chama o *plugin* de simulação que irá utilizá-los e retornará o resultado da simulação para o usuário.

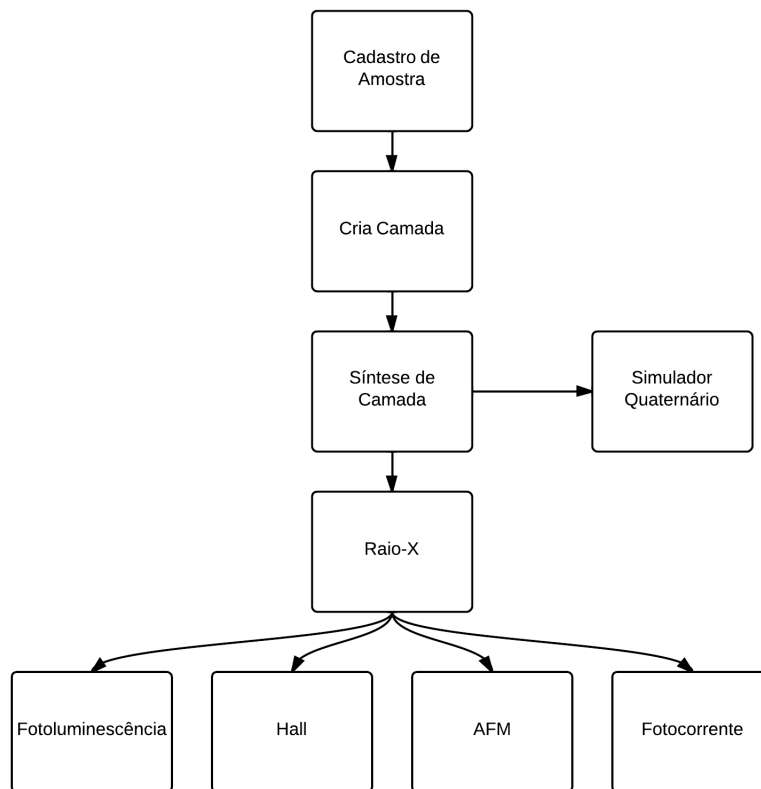


**Figura 5.8.** Resultado da rede neural nas etapas de treinamento, validação e teste para a saída de Desvio Padrão.

## 5.2 Laboratório de Semicondutores (PUC-Rio)

O *Laboratório de Semicondutores da PUC-Rio*, ou simplesmente *LabSem*, é um laboratório de nanotecnologia integrado ao Grupo de Optoeletrônica do Centro de Estudos em Telecomunicações da PUC-Rio. No *LabSem* são realizadas pesquisas básicas e aplicada na área de materiais semicondutores III-V para dispositivos eletrônicos e optoeletrônicos utilizados em telecomunicações. As atividades abrangem desde o crescimento epitaxial (por camadas) até a confecção dos dispositivos, passando por diferentes etapas de estudo e caracterização de propriedades ópticas, elétricas, eletrônicas, optoeletrônicas e estruturais [LabSem, 2013].

A construção de um sistema para gerenciar as informações deste laboratório possui mais de um objetivo, o primeiro é suprir as necessidades dos pesquisadores com um sistema computacional capaz de controlar os experimentos realizados no *LabSem*. Outro objetivo é demonstrar que o *NanoTrack* é capaz de atender um laboratório de



**Figura 5.9.** Workflow do Laboratório de Semicondutores (PUC-Rio).

nanotecnologia com problemas reais encontrados no dia-a-dia do pesquisador. Por último, prover um ambiente de simulação que seja capaz de obter resultados satisfatórios para o pesquisador através da integração de *plugins* ao LIMS.

### 5.2.1 Modelagem

O processo de modelagem do laboratório *LabSem* foi iniciado em uma visita ao laboratório, onde foi realizada uma reunião com os pesquisadores. Durante a visita foi criado um esboço, mostrado na Figura 4.2 do capítulo 4. Em posse deste esboço, o *workflow* foi passado para um modelo digital com o intuito de documentar a estrutura do *LabSem*, este pode ser visto na Figura 5.9. Este *workflow*, devido a existência de um número maior de atividades e atributos, possui uma complexidade maior que o *workflow* de *Pontos Quânticos*.

Foram identificados oito processos oriundos do esboço elaborado junto aos pesquisadores do *LabSem*. Para cada processo foi criada uma atividade no *workflow* que o representasse. Além destas, também foi criada uma atividade específica para o simulador desenvolvido e encapsulado em um *plugin*. Para facilitar o entendimento das

atividades, é possível classificá-las em grupos. Veja esta classificação na tabela 5.5.

<i>Grupo</i>	<i>Atividade</i>
Novo experimento	Cadastro de Amostra
Camada	Cria Camada
Síntese	Síntese de Camada
Caracterização Obrigatória	Raio-X
Caracterização Opcional	Fotoluminescência
	Hall
	AFM
	Fotocorrente
Plugin	Simulador Quaternário

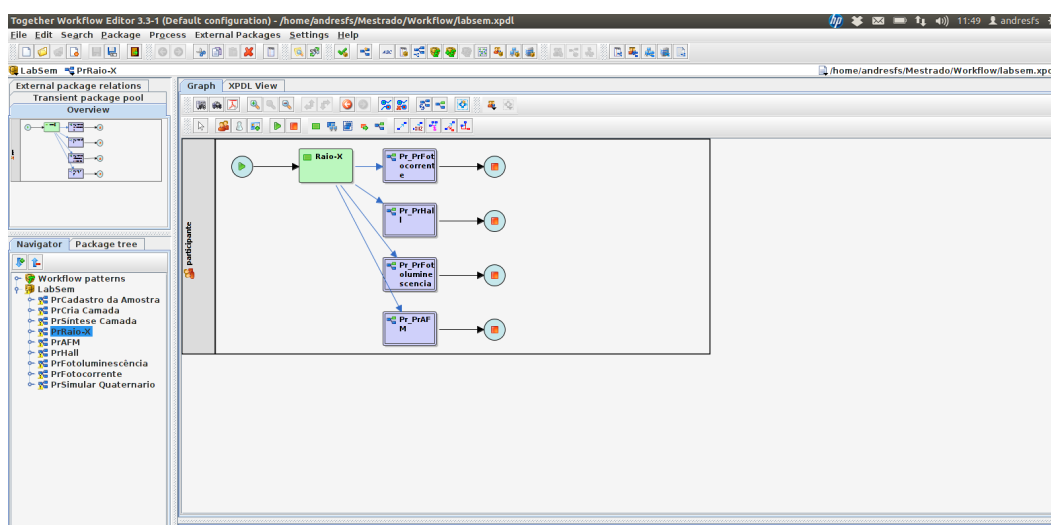
**Tabela 5.5.** Lista de atividades identificadas durante a criação do workflow do Laboratório de Semicondutores.

Basicamente, sempre que o pesquisador iniciar um novo experimento, o primeiro passo é realizar o cadastro da amostra. Uma vez que a amostra esteja cadastrada o próximo passo é criar as camadas desta amostra. As camadas podem ser classificadas em três tipos: *Substrato*, *Poço Quântico* e *Ponto Quântico*. A próxima etapa é a atividade de Síntese. Esta etapa depende do tipo da camada escolhido na etapa anterior, pois dependendo da escolha, alguns atributos não precisarão ser preenchidos. Além dos atributos específicos por camada, existem atributos que são comuns a todos os tipos. A atividade de Raio-X é uma caracterização obrigatória, realizada sempre após a atividade de síntese e apenas após a sua execução que as demais caracterizações ficam disponíveis para que o pesquisador as execute. As caracterizações opcionais não ocorrem sempre, apenas quando surge a necessidade de executá-las.

Para que o *NanoTrack* consiga interpretar o *workflow* é necessário codificá-lo para XPDL. Para isto, foi utilizado a ferramenta *Together Workflow Editor*, onde foram criadas todas as atividades e seus atributos no *workflow*. Uma etapa desta codificação pode ser visualizada na Figura 5.10.

Cada atividade é composta por diversos atributos que representam suas propriedades. Estas deverão ser armazenadas no banco de dados com o intuito de cadastro e possibilitar futuras consultas. As propriedades de cada atividade foram obtidas através dos atributos disponíveis em uma planilha eletrônica, que era utilizada como cadastro de experimentos pelos pesquisadores do *LabSem*. Veja na Figura 5.11 a interface do TWE com os atributos que compõem a atividade de síntese do *workflow*.

Após o *workflow*, já em seu formato XPDL, ser carregado no *NanoTrack*, este fica disponível para que os pesquisadores possam cadastrar seus experimentos. Na Figura 5.12 é possível visualizar um experimento com a atividade de “Síntese de Camada”



**Figura 5.10.** Interface do Together Workflow Editor durante a codificação da atividade de Raio-X no workflow do LabSem.

The screenshot shows a window titled 'Workflow variables, 80 element(s)'. It contains a table with the following columns: Id, Name, Data type, and Initial value. The table lists 80 variables, all with a 'Basic type -> Float' data type.

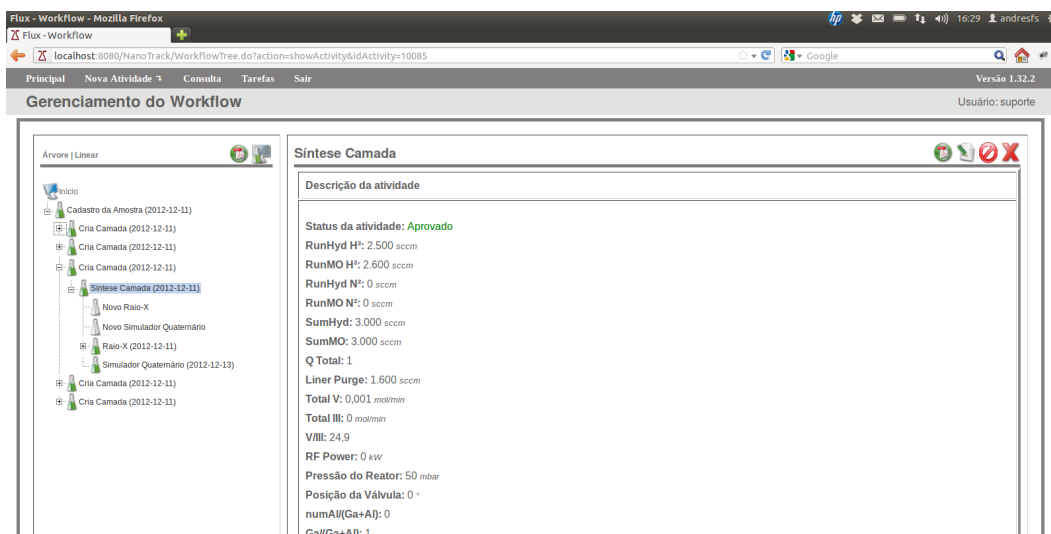
Id	Name	Data type	Initial value
numRunHydH2	RunHyd H <sup>2</sup>	Basic type -> Float	
numRunMOH2	RunMO H <sup>2</sup>	Basic type -> Float	
numRunHydN2	RunHyd N <sup>2</sup>	Basic type -> Float	
numRunMON2	RunMO N <sup>2</sup>	Basic type -> Float	
numSumHyd	SumHyd	Basic type -> Float	
numSumMO	SumMO	Basic type -> Float	
numQTotal	Q Total	Basic type -> Float	
numLinerPurge	Liner Purge	Basic type -> Float	
numTotalV	Total V	Basic type -> Float	
numTotalIII	Total III	Basic type -> Float	
numV_III	V/III	Basic type -> Float	
numRFPower	RF Power	Basic type -> Float	
numPressaoRea...	Pressão do Reator	Basic type -> Float	
numPosicaoValv...	Posição da Válvula	Basic type -> Float	
numAl_Ga_Al	numAl/(Ga+Al)	Basic type -> Float	
numGa_Ga_Al	Ga/(Ga+Al)	Basic type -> Float	
numGa_Ga_In	Ga/(Ga+In)	Basic type -> Float	
numAl_Ga_Al_In	Al/(Ga+Al+In)	Basic type -> Float	
numGa_Ga_Al_In	Ga/(Ga+Al+In)	Basic type -> Float	

**Figura 5.11.** Oitenta atributos que compõem a atividade de síntese do workflow de LabSem.

e seus atributos já carregados. Também é possível visualizar, no lado esquerdo da imagem, que esta amostra possui 5 camadas cadastradas além de outras atividades que já foram executadas.

### 5.2.1.1 Tabelas Externas

Os atributos de todas as atividades deste *workflow*, com exceção da atividade do simulador, foram marcados para utilizarem tabelas auxiliares durante a codificação em XPD. Para cada uma destas atividades foi criada uma tabela responsável por armazenar os dados de sua respectiva atividade. As tabelas externas criadas e suas respectivas atividades estão mostradas na tabela 5.6.



**Figura 5.12.** Interface do Nanotrack durante a atividade de síntese de camada do workflow do LabSem.

<i>Atividade</i>	<i>Tabela</i>
Cadastro de Amostra	TB_LABSEM_CADASTRO_AMOSTRA
Cria Camada	TB_LABSEM_CRIA_CAMADA
Síntese de Camada	TB_LABSEM_SINTESE
Raio-X	TB_LABSEM_RAIOX
Fotoluminescência	TB_LABSEM_FOTOLUMINESCENCIA
Hall	TB_LABSEM_HALL
AFM	TB_LABSEM_AFM
Fotocorrente	TB_LABSEM_FOTOCORRENTE

**Tabela 5.6.** Tabelas externas criadas para as atividades do workflow do LabSem

## 5.2.2 Importação de dados

O *LabSem* arquivava os experimentos realizados no laboratório em planilhas eletrônicas. Estes registros foram migrados para o banco de dados do LIMS. Devido ao grande número de experimentos, a importação destes dados foi realizada através de um *script*. Este *script* inseriu os dados dos experimentos em todas as tabelas do banco de dados das atividades que possuía algum registro na planilha eletrônica. Este procedimento, se fosse realizado de modo manual, iria demandar bastante tempo do usuário, além de estar propício à inserção de erros devido a diversos fatores humanos.

Além de inserir os dados dos experimentos existentes, o *NanoTrack* precisa que os experimentos sejam criados nas tabelas de sua estrutura interna. Logo, para evitar que a ferramenta não consiga utilizar estes experimentos posteriormente, foi necessário gerar os registros de controle interno da plataforma Flux® também. Desta forma, o



*NanoTrack* não terá problema ao manipulá-los.

### 5.2.3 Ambiente de simulação

O ambiente de simulação desenvolvido para o *workflow* do *LabSem* é um pouco mais complexo do que o desenvolvido no estudo de caso anterior. Neste, a rede neural desenvolvida, além de ser capaz de gerar resultados para duas saídas simultaneamente, é utilizada de maneira indireta e o simulador é baseado em um algoritmo genético para otimizar os parâmetros de síntese dos quaternários. Os quaternários são materiais semicondutores compostos por quatro elementos. No *LabSem*, normalmente eles utilizam: Alumínio, Índio, Gálio e Arsênio, formando um composto conhecido como *AlInGaAs*.

Como já foi detalhado nos capítulos anteriores, uma rede neural funciona da seguinte forma: uma série de parâmetros de entrada é passada para ela, então, estes dados são processados pela rede neural que irá devolver os resultados. No entanto, o propósito do simulador desenvolvido para este *workflow* é outro. É desejado descobrir quais os parâmetros de entrada que irão resultar nas saídas desejadas, ou seja, o pesquisador irá fixar o resultado desejado e um algoritmo (algoritmo genético) irá buscar, através da utilização da rede neural, um conjunto de entradas que resultará nos valores desejados pelo usuário.

#### 5.2.3.1 Treinamento da Rede Neural Artificial

A rede neural para simular experimentos realizados com quaternários do *LabSem* foi criada através da utilização da biblioteca de funções que está disponível no *Matlab*, como explicado na seção 4.3.3. Os parâmetros de entrada e saída necessários para o treinamento da rede neural foram identificados pelos pesquisadores do *LabSem*, que também enviaram, através de uma planilha eletrônica, os dados de experimentos somente com os parâmetros de treinamento necessários.

Durante a etapa de síntese do quaternário, o reator é ajustado e, de acordo com esta regulagem, os nanomateriais são criados com concentrações diferentes de *Índio*, *Gálio* e *Alumínio*. A rede neural desenvolvida neste trabalho visa simular, com base nos parâmetros de ajuste do reator, as concentrações destes elementos nos quaternários. Veja na tabela 5.7 os parâmetros de entrada (configuração do reator) e saída (concentrações dos elementos) identificados pelos pesquisadores do *LabSem*.

Baseado na planilha eletrônica disponibilizada pelos pesquisadores, um arquivo texto é gerado com todos os experimentos. O arquivo gerado é então carregado para a memória do *Matlab* e as colunas referentes as entradas e saídas são mapeadas. Esta rede

<i>Atributo</i>	<i>Classificação</i>	<i>Atividade</i>
AsH <sub>3</sub> (mol/min)	Entrada	Síntese
TMIn (mol/min)		
TMGa (mol/min)		
TMAI (mol/min)		
Temperatura - Pressão (°C - mbar)		
% Índio	Saída	Raio-X
% Gálio		
% Alumínio		

**Tabela 5.7.** Lista de atributos de entrada e saída utilizados na rede neural.

neural foi configurada para encontrar resultados de concentração de Gálio e Alumínio num mesmo processamento, ou seja, ela tentará durante sua execução, ajustar os pesos de sua estrutura interna para encontrar duas saídas simultaneamente.

Em seguida, é realizado a normalização dos dados para evitar que grandes variações de valores dos experimentos causem problemas nos resultados da rede neural. Com os dados normalizados, eles são separados em grupos para as etapas de treinamento e teste da rede neural. O agrupamento de treinamento possui 80% dos experimentos, sendo que destes, 20% são utilizados na validação. Os demais 20% são utilizados para testar a rede neural.

Uma vez que o agrupamento dos dados é realizado, a rede neural está pronta para iniciar a etapa de aprendizado. Durante esta etapa, são verificadas várias configurações da rede neural onde são alteradas a quantidade de camadas internas e a quantidade neurônios por camada. Cada rede neural treinada é comparada com a melhor rede neural obtida até aquele momento e se a nova rede neural for melhor que todas as demais redes neurais ela é armazenada. Ao final da etapa de aprendizado a melhor rede neural treinada estará disponível e é guardada em um arquivo para que seja possível utilizá-la em um momento futuro.

### 5.2.3.2 Utilizando a Rede Neural Artificial

Ao final da etapa de treinamento, a melhor rede neural obtida é gravada em um arquivo e será utilizada nesta etapa. Primeiro, é necessário carregá-la para a memória do *Matlab*. Após este carregamento, todo o ambiente que existia no momento que ela foi gravada fica disponível para ser utilizado.

Dado um conjunto de dados informado pelo pesquisador, composto pelos parâmetros de entrada apresentados na tabela 5.7, a rede neural irá inferir a concentração de *Gálio* e *Alumínio* que será resultante deste experimento. Para isto, é necessário

que os dados informados sejam normalizados seguindo o mesmo padrão adotado durante a etapa de treinamento da rede neural, a fim de que eles não sejam distorcidos. Para utilizar o mesmo padrão de normalização, basta carregar juntamente com os dados informados pelo pesquisador os dados utilizados durante o treinamento e que está disponível no ambiente do *Matlab* após o carregamento da melhor rede neural.

Após a normalização do dados, os dados informados pelo pesquisador é passado para a rede neural que irá processá-los e ao final retornar os resultados obtidos para a concentração de *Gálio* e *Alumínio*. Em seguida, aplica-se a estes resultados, uma função para reverter a normalização feita anteriormente e obter os valores reais. Para descobrir a concentração de *Índio*, basta fazer a subtração de 100 da soma das concentrações de *Gálio* e *Alumínio*.

### 5.2.3.3 Resultados da Rede Neural

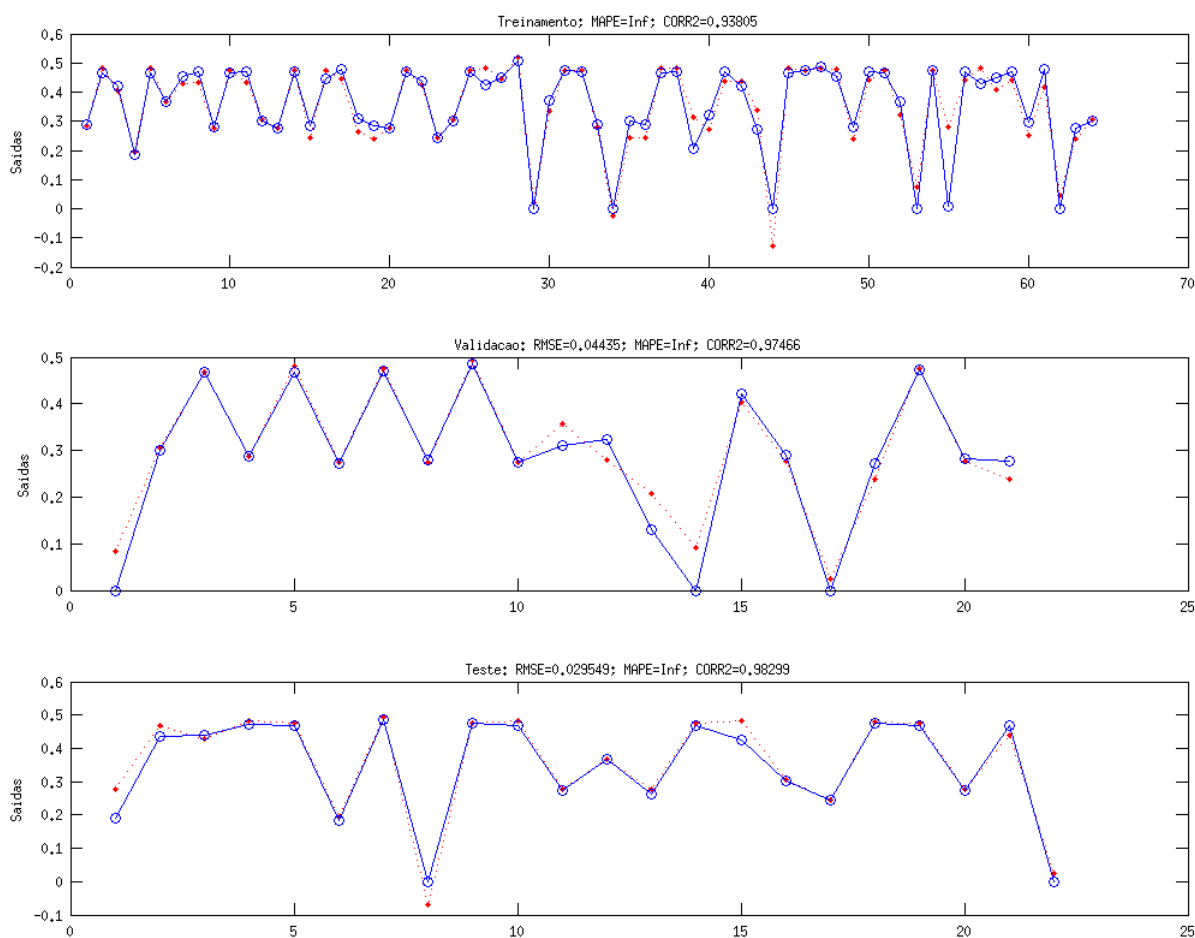
Para este ambiente de simulação foi necessário apenas o treinamento de uma única rede neural. Ela é capaz de inferir as concentrações dos elementos *Gálio* e *Alumínio* e calcular com base nos resultados a concentração de *Índio*. Esta rede neural treinada, além de possuir as camadas de entrada e saída, possui uma única camada escondida contendo seis neurônios.

Embora seja uma única rede, foram gerados gráficos com os resultados obtidos separadamente, a fim de demonstrar com maior clareza e precisão os resultados tanto para a inferência de concentração de *Gálio*, quanto para a concentração de *Alumínio*.

Nas figuras a seguir serão mostrados três gráficos em cada uma delas. Estes gráficos referem-se as etapas de treinamento, validação e teste. A linha de cor vermelha (tracejada - ponto) representa o valor obtido pela rede neural e a linha de cor azul (contínua - bolinha) representa o valor que era esperado.

Os resultados obtidos com a rede neural para inferência de concentração de *Gálio* foram excelentes. Durante a etapa de testes da rede neural, o erro médio obtido foi abaixo de 3% que é um resultado bastante satisfatório. Além disto, o coeficiente de correlação foi de 0,983, que é muito próximo de 1 e indica que a rede neural está aproximando todos os seus resultados aos resultados esperados. Veja na Figura 5.13 os três gráficos das etapas de treinamento, validação e teste.

A mesma rede neural infere valores para a concentração de *Alumínio* e os resultados foram muito bons. Embora os resultados com a obtenção da concentração de *Gálio* tenha sido melhores, a diferença do erro médio é praticamente irrisória. Para a inferência de concentração de *Alumínio*, o erro médio da rede neural foi de aproximadamente 3,5% e o coeficiente de correlação foi de 0,972, que também é muito próximo



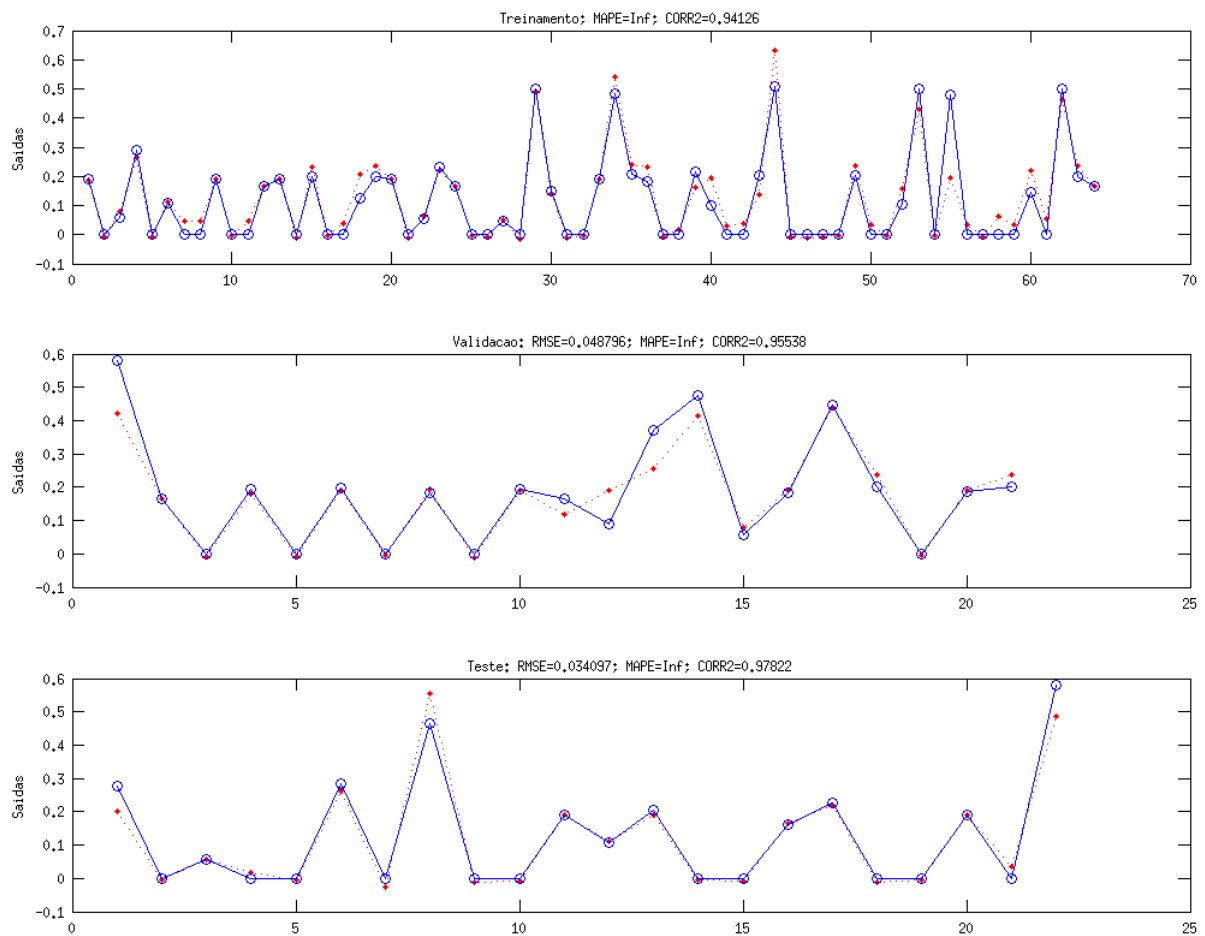
**Figura 5.13.** Resultado da rede neural nas etapas de treinamento, validação e teste para a concentração de Gálio.

de 1 e demonstra que os resultados da rede neural estão muito bons. Os resultados obtidos nas etapas de treinamento, validação e teste podem ser vistos na Figura 5.14.

#### 5.2.3.4 Quaternário reverso

A utilização da rede neural consiste no usuário informar os dados de entrada e, com base nestes dados ela irá gerar os dados de saída. O *Quaternário Reverso* é um algoritmo criado para fazer o processo inverso, ou seja, o usuário irá informar as concentrações de *Índio*, *Gálio* e *Alumínio* que ele deseja e o algoritmo irá gerar os parâmetros que deverão ser utilizado como entrada no reator para que as concentrações desejadas sejam alcançadas.

Para o desenvolvimento desta funcionalidade foi utilizado um algoritmo genético que está disponível na biblioteca de funções do *Matlab*. Para o seu funcionamento, o algoritmo genético precisa que alguns parâmetros de configuração sejam ajustados.



**Figura 5.14.** Resultado da rede neural nas etapas de treinamento, validação e teste para a concentração de Alumínio.

Primeiro é necessário definir os valores máximos e mínimos assumidos por cada um dos parâmetros de entrada, estes valores irão definir um intervalo de busca e dará suporte ao algoritmo genético durante sua execução. Em seguida, é preciso definir o tamanho da população que será gerada, bem como o número de gerações. Além disto, é necessário definir a função de *fitness* que será utilizada para avaliar os indivíduos gerados. A função de *fitness* é de fundamental importância para que o algoritmo obtenha resultados satisfatórios e é neste momento que é realizada a integração com o a rede neural desenvolvida.

Um indivíduo, em algoritmos genéticos, é uma possível solução para o problema. Em nosso problema, o indivíduo é composto por cinco atributos que são exatamente os atributos classificados como “Entrada” e estão mostrados na tabela 5.7. Para que ele sempre crie indivíduos válidos, o algoritmo genético utiliza a informação dos intervalos de valores que foram informados em sua configuração. Desta forma, mesmo que o indivíduo gere um resultado que não atende ao problema, ele será válido.

A rede neural, utilizada através da função de *fitness*, é que irá determinar se o indivíduo é bom ou ruim. O seu funcionamento é similar ao de uma função, onde cada um dos atributos que compõem o indivíduo é uma variável e estas são aplicadas à rede neural que retornará um valor. Então este valor é comparado com o resultado esperado. Quanto mais próximo o valor obtido estiver do desejado, melhor é o indivíduo. Com base na qualidade do indivíduo e das propriedades evolutivas do algoritmo genético, ele consegue gerar indivíduos melhores. Além disto, o algoritmo consegue explorar uma grande parte do espaço de busca. Assim, ao final de sua execução ele seleciona os indivíduos com os melhores resultados como possíveis soluções para o problema.

Com o objetivo de atender a uma solicitação dos pesquisadores do *LabSem* e também de validar a capacidade do algoritmo em obter resultados, alguns testes foram realizados. Segundo os pesquisadores do *LabSem*, existe uma carência de documentos na literatura que mostram o processo de síntese de quaternários para algumas composições, como por exemplo, as listadas na tabela 5.8. Segundo eles, os parâmetros de entrada que resultarão nestas composições não estão documentados, o que dificulta a criação destes nanomateriais com estas composições.

	<i>Índio</i>	<i>Gálio</i>	<i>Alumínio</i>
Problema 1	50,0	30,0	20,0
Problema 2	50,0	32,0	18,0

**Tabela 5.8.** Composições de quaternários carentes de documentação.

Para verificar a capacidade de obtenção de resultados, o algoritmo foi testado buscando encontrar os parâmetros de entrada que iriam resultar nas composições desejadas. Veja na Figura 5.15 como foi o processo de convergência até a obtenção do resultado final.

A Figura 5.15 foi obtida durante a execução para a obtenção dos resultados para a composição: *Índio* - 50%, *Gálio* - 32% e *Alumínio* - 18%. Os parâmetros de entrada obtidos, bem como as composições estão disponíveis na coluna “Solução 2” mostrada na tabela 5.9. Na coluna “Solução 1” estão os resultados dos parâmetros obtidos para a composição: *Índio* - 50%, *Gálio* - 30% e *Alumínio* - 20%. A coluna “Solução 1” representa a solução para o “Problema 1” da tabela 5.8, assim como a “Solução 2” representa a solução para o “Problema 2”.

Se observarmos as composições obtidas através do algoritmo genético podemos perceber que o erro das composições estão na segunda e terceira casas decimais, o que demonstra uma precisão muito boa. Estes valores foram passados para a equipe do *LabSem* para que estas amostras sejam sintetizadas e possibilitar uma avaliação

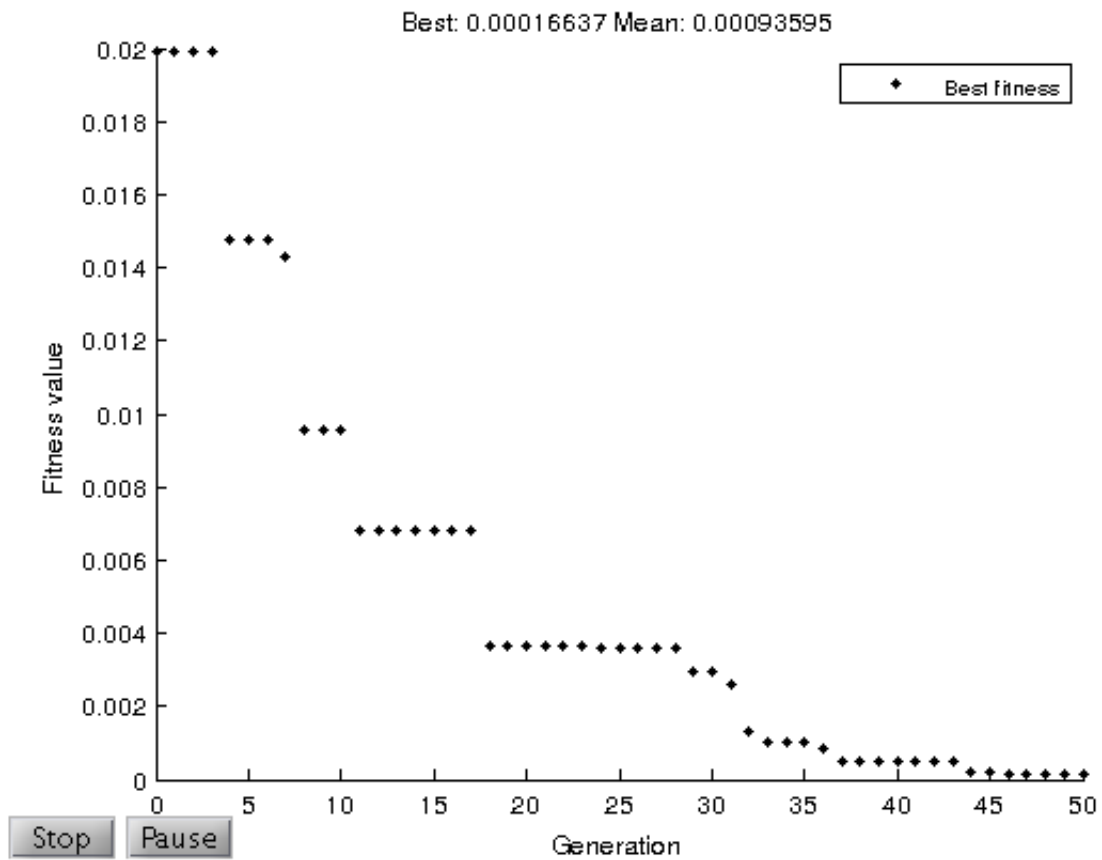
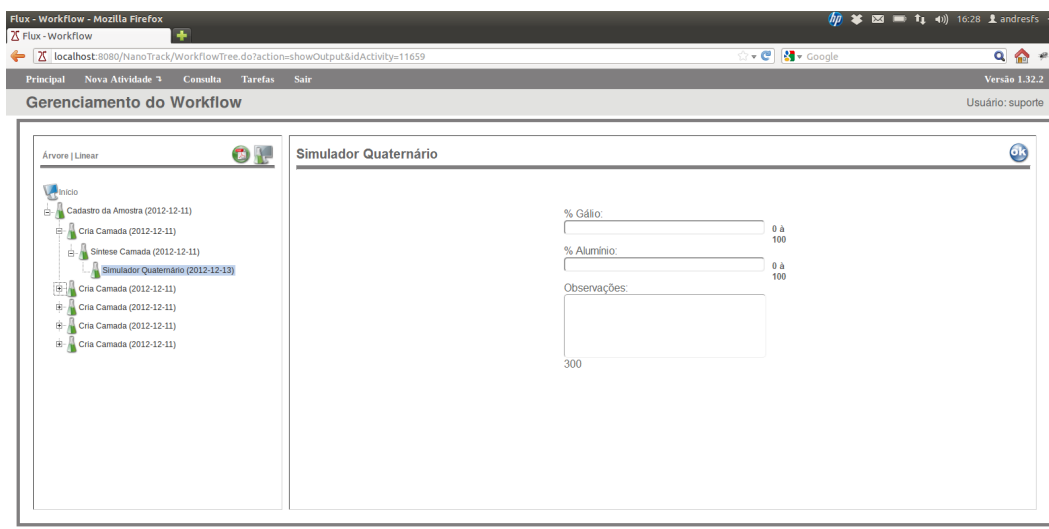


Figura 5.15. Indivíduo evoluiu durante a execução do algoritmo genético.

Atributos	Solução 1	Solução 2
AsH3	0,00106967405496527000	0,00103590058224384000
TMIIn	0,00003358172678221790	0,00002833019879423890
TMGa	0,00001605618502373020	0,00001616705082262070
TMAI	0,00000822217555734401	0,00001109526104758810
Temperatura	600	650
Pressão	50	100
% Índio	49,98	50,02
% Gálio	30,02	31,98
% Alumínio	20,00	18,00

Tabela 5.9. Bons resultados obtidos com algoritmo genético.

completa sobre a precisão do algoritmo, mas elas ainda não foram produzidas. Mas já é possível afirmar que, computacionalmente, os resultados estão satisfatórios.



**Figura 5.16.** Interface do “Simulador Quaternário” no NanoTrack durante o preenchimento dos parâmetros de entrada do simulador.

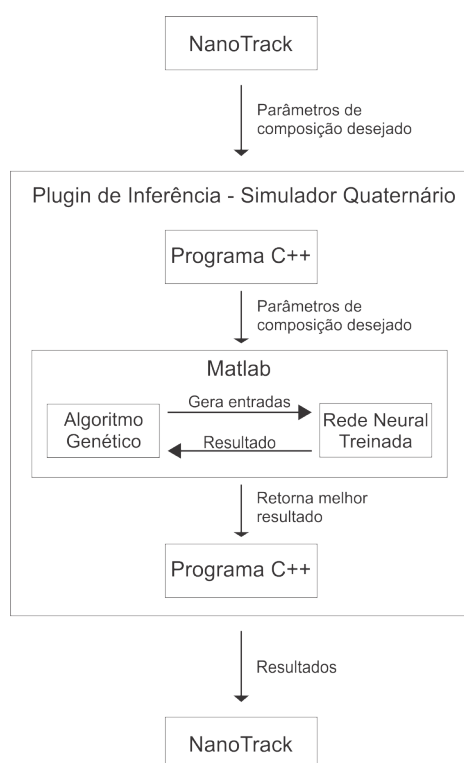
### 5.2.3.5 Integração com plugin

Neste estudo de caso apenas o “Quaternário Reverso” foi encapsulado em um *plugin*, sendo que a utilização da rede neural desenvolvida é feita durante a operação do “Quaternário Reverso”. O primeiro passo para a integração do *plugin* com o *NanoTrack* foi a criação da atividade que irá representá-lo no *workflow* do *LabSem*, esta atividade foi nomeada de “Simulador Quaternário”.

A atividade “Simulador Quaternário” é composta pelos atributos: % *Gálio*, % *Alumínio* e *Observações*, como pode ser visto na Figura 5.16. No momento em que o *plugin* é acionado, os atributo % *Gálio* e % *Alumínio* são passados como parâmetros para ele. Estes valores são as concentrações de Gálio e Alumínio desejadas pelo pesquisador. O *plugin* irá buscar o melhor conjunto de valores para os atributos de configuração do reator (AsH<sub>3</sub>, TMI<sub>n</sub>, TMGa, TMAI, Temperatura e Pressão) que aproximará dos valores desejados pelo pesquisador. Ao final de sua execução os resultados obtidos para as concentrações são exibidos para o usuário, bem como o conjunto de entrada que gerará este resultado.

A Figura 5.17 mostra como é o funcionamento e as interações que existem durante o funcionamento do *plugin*. Os atributos da atividade do *NanoTrack* são repassados para o *plugin* que na sequência aciona o “Quaternário Reverso” que é executado dentro do *Matlab*. Durante a execução do “Quaternário Reverso”, este interage com a rede neural desenvolvida, a fim de avaliar a qualidade dos resultados obtidos. Por meio destes resultados parciais e através das propriedades evolutivas do algoritmo genético, ele consegue melhorar a solução. Ao final, a melhor solução é exibida para o usuário.





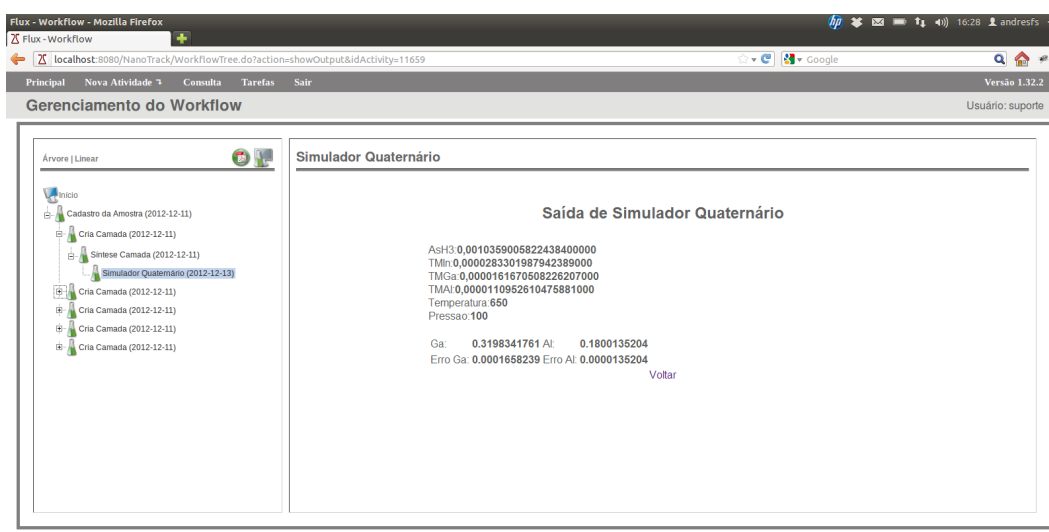
**Figura 5.17.** Estrutura do plugin mostrando as interações entre o NanoTrack, plugin e o Matlab.

Veja na Figura 5.18 o simulador desenvolvido para este *workflow*, integrado ao *NanoTrack* através da atividade “Simulador Quaternário”, com o resultado de uma simulação realizada por ele.

### 5.3 Nanotubos de Carbono (Física-UFMG)

O *Laboratório de Nanomateriais*, que pertence ao Departamento de Física da UFMG, tem como principais áreas de pesquisa a síntese e aplicação de diversos tipos de materiais nanoestruturados, em particular, nanotubos de carbono de parede única e múltiplas paredes. Os nanotubos de carbono são estruturas cilíndricas formadas por átomos de carbono, cujo diâmetro é de um a três nanômetros (nm). As propriedades encontradas nestes nanotubos de carbono fazem deles nanoestruturas muito promissoras para a nanotecnologia. Eles são extremamente resistentes, muito leves e, quanto às suas propriedades eletrônicas, podem ser condutor ou semicondutor, dependendo da forma como os átomos estão dispostos, [Laboratório de Nanomateriais, 2013].

Para auxiliar na organização das informações geradas no *Laboratório de Nanomateriais*, um *workflow* foi construído com o objetivo de inseri-lo no sistema *NanoTrack*



**Figura 5.18.** Ambiente de simulação do workflow representado pela atividade “Simulador Quaternário”

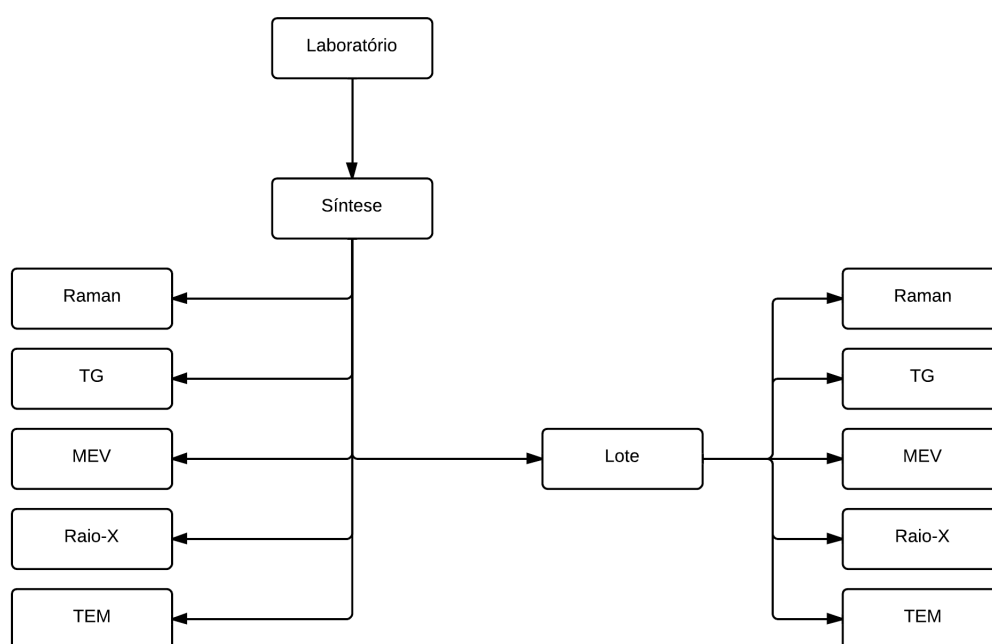
e possibilitar que os pesquisadores deste laboratório usufruam de todas os benefícios que um LIMS pode lhes proporcionar. Nenhum ambiente de simulação foi incorporado a este *workflow*. Entretanto, é possível, em um momento futuro, que algum simulador seja desenvolvido e integrado ao *workflow* e disponibilizado na ferramenta.

### 5.3.1 Modelagem

O *workflow* desenvolvido neste estudo de caso representa o processo de criação dos *Nanotubos de Carbono*. Ele foi elaborado após reuniões realizadas com os pesquisadores do laboratório. A estrutura final deste *workflow* pode ser visualizada na Figura 5.19. Este *workflow* possui características interessantes que não haviam sido exploradas nos estudos de casos anteriores, mas que foram implantadas com sucesso.

Foram identificados doze processos que podem ocorrer durante a experimentação com *Nanotubos de Carbono*, sendo que alguns destes processos não ocorrem sempre, portanto, são alternativos. Para cada processo identificado, uma atividade foi criada no *workflow* para representá-lo. Além destas, foi criada uma outra atividade, identificada como “Laboratório”, que visa possibilitar a criação de lotes de nanoestruturas contendo diferentes amostras. O *NanoTrack* apenas possibilita a junção de amostras quando elas estão na mesma hierarquia e com a criação desta atividade, todas as amostras ficam com uma atividade em comum e, portanto, podem fazer parte de um mesmo lote. Veja na tabela 5.10 as atividades que foram identificadas e criadas neste *workflow*.

O processo experimental inicia com o cadastramento do Laboratório no LIMS, este procedimento é realizado apenas uma vez. Assim, todas as demais atividades ca-



**Figura 5.19.** Workflow do Laboratório de Nanotubos de Carbono.

<i>Grupo</i>	<i>Atividade</i>
Laboratório	Laboratório
Síntese	Síntese
Caracterização Opcional	Raman
	TG
	MEV
	Raio-x
	TEM
Lote	Lote
Caracterização Opcional do Lote	Raman
	TG
	MEV
	Raio-x
	TEM

**Tabela 5.10.** Lista de atividades identificadas durante a criação do workflow de Nanotubos de Carbono.

dastradas neste *workflow* serão “atividades-filha” da atividade “Laboratório”. Uma vez que o cadastro do laboratório já tenha sido feito, a atividade de síntese fica disponível para ser realizada.

Na etapa de síntese é feito o cadastramento, no LIMS, da amostra de um determinado experimento. Esta etapa possui características peculiares que ainda não haviam sido implementadas nos estudos de caso anteriores. Após o cadastro da amostra com os

dados de síntese é necessário que o usuário forneça ao sistema informações resultantes da síntese, que são: a massa final e o rendimento. Somente após o fornecimento destes dados que as atividades posteriores ficaram disponíveis para o usuário.

Toda síntese, no final, vira um lote. Mesmo que este lote contenha apenas uma única síntese. Em geral, o lote é a atividade responsável por agrupar dados de várias síntese em apenas um registro, simulando a criação de um lote de produção industrial. Durante a atividade de criação do lote, as amostras cujo resultado de síntese já tenham sido informados estarão disponíveis e o usuário poderá selecioná-las. Estas amostras selecionadas é que irão compor o lote. As atividades de caracterização, tanto da amostra individualmente quanto do lote, são opcionais. Além disto, a execução da caracterização de uma amostra individualmente não impede que o lote desta amostra também seja caracterizado.

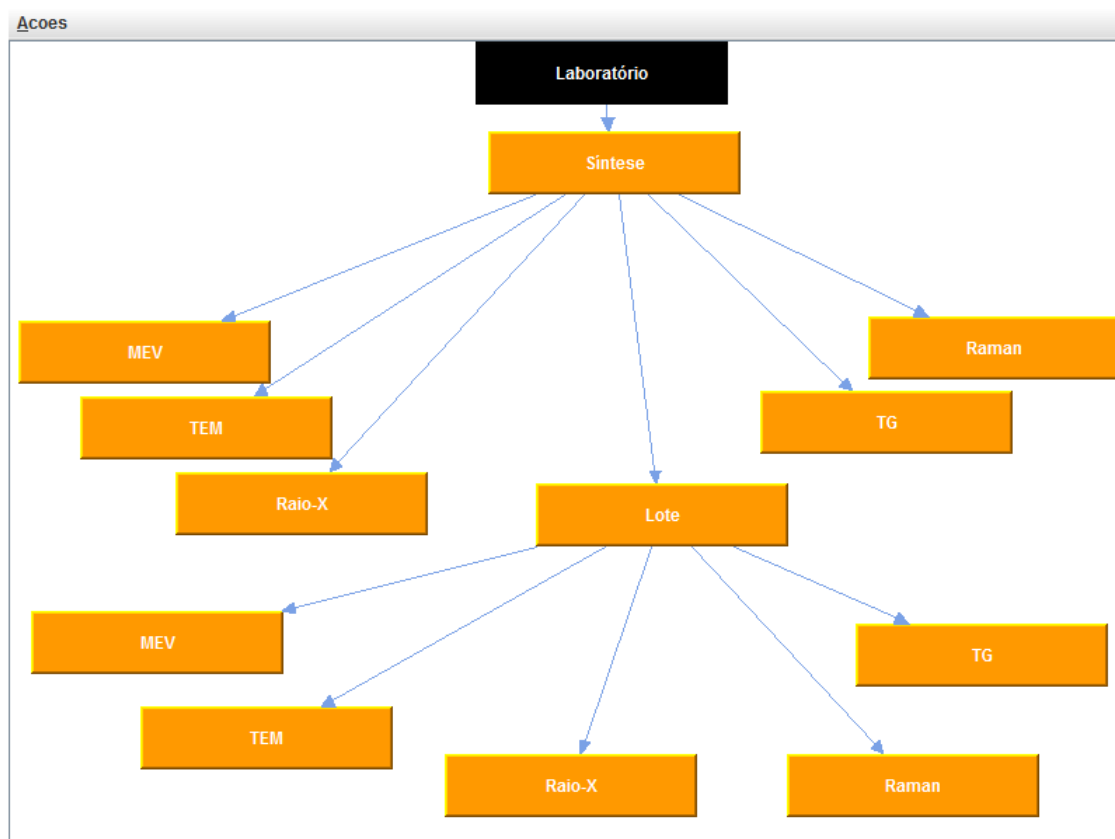
Depois de modelado, o *workflow* foi transcrito em XPDL para que o *NanoTrack* pudesse interpretá-lo. Diferentemente dos outros dois casos de estudo que havia sido codificado pelo *Together Workflow Editor*, neste *workflow* foi utilizado o *Fluxar*. Veja na Figura 5.20 a interface da ferramenta após a conclusão da codificação do *workflow*. O *Fluxar* é um software em desenvolvimento pelo laboratório LUAR, mas que já possui as principais funcionalidades implementadas. A utilização deste software para tal codificação teve objetivo de testar o software, através da codificação de um *workflow* por usuários que não trabalharam no desenvolvimento do *Fluxar* e transmitir a experiência, bem como sugestões e críticas aos membros do LUAR.

As atividades são compostas por atributos que representam as propriedades dos processos. Durante o cadastramento das atividades, estes atributos são preenchidos pelos usuários e, ao final, são armazenados no banco de dados. Durante as reuniões realizadas para o mapeamento do laboratório, as propriedades de cada processo foram repassadas e anotadas para a implementação. Veja na Figura 5.21, a lista de atributos que compõem a atividade de síntese deste *workflow*.

O resultado deste *workflow* implementado no *NanoTrack* pode ser visto na Figura 5.22, que mostra uma atividade de síntese durante a etapa de cadastro. Os atributos mostrados na Figura 5.21 estão presentes nesta interface, exceto massa final e rendimento que serão preenchidos após o cadastro como resultado da síntese.

### 5.3.1.1 Tabelas Externas

Durante o processo de codificação deste *workflow* em XPDL, os atributos de todas as atividades foram marcadas para utilizarem tabelas externas. Para cada atividade do *workflow*, uma tabela foi criada para armazenar os dados acerca da atividade. Veja na



**Figura 5.20.** Interface do Fluxar com o workflow de Nanotubos de Carbono sendo codificado para XPDL.

tabela 5.11 a lista de tabelas externas criadas e suas respectivas atividades.

<i>Atividade</i>	<i>Tabela</i>
Laboratório	TB_NANOTUBOS_LABORATORIO
Síntese	TB_NANOTUBOS_SINTESE
Raman	TB_NANOTUBOS_RAMAN
TG	TB_NANOTUBOS_TG
MEV	TB_NANOTUBOS_MEV
Raio-X	TB_NANOTUBOS_RAIOX
TEM	TB_NANOTUBOS_TEM
Lote	TB_NANOTUBOS_LOTE
Raman	TB_NANOTUBOS_LOTERAMAN
TG	TB_NANOTUBOS_LOTETG
MEV	TB_NANOTUBOS_LOTEMEV
Raio-X	TB_NANOTUBOS_LOTERAIOX
TEM	TB_NANOTUBOS_LOTETEM

**Tabela 5.11.** Tabelas externas criadas para as atividades do workflow de Nanotubos de Carbono

id	Nome	Tipo
Criar	Novo	Atributo
numMassaInicial	Massa Inicial	Decimal
dscGasArraste	Gás de Arraste	Texto
numFluxoArraste	Fluxo de Arraste	Decimal
dscFonteCarbono	Fonte de Carbono	Texto
dscTipoBarco	Tipo de Barco	Texto
dscCatalisador	Catalisador	Texto
numTemperaturaCresci...	Temperatura de Crescim...	Decimal
numTempoCrescimento	Tempo de Crescimento	Inteiro
numFluxoFonteCarbono	Fluxo da Fonte de Carbo...	Decimal
output	output	Texto
numRendimento	Rendimento	Decimal
numMassaFinal	Massa Final	Decimal

**Figura 5.21.** Atributos da atividade de síntese do workflow de Nanotubos de Carbono.

No capítulo 4 foi justificado que a utilização de tabelas externas era devido a facilidade que estas trariam na utilização dos *plugins*. Mesmo que nenhum *plugin* tenha sido incorporado a este *workflow*, a possibilidade de uma futura incorporação justifica o desenvolvimento das atividades com as tabelas externas. Desta forma, não será necessário fazer nenhuma alteração na estrutura do *workflow* e nem migrar os dados para novas tabelas, uma vez que estes já estarão armazenados em tabelas externas.

The screenshot displays a web browser window titled 'Flux - Workflow - Mozilla Firefox'. The address bar shows the URL: `localhost:8080/NanoTrack/WorkflowTree.do?updateActivity=null&action=executeActivity&instanceId=2402&fatherActivityId=11669&nextActivityName=F`. The browser's navigation bar includes 'Principal', 'Nova Atividade', 'Consulta', 'Tarefas', and 'Sair'. The page title is 'Gerenciamento do Workflow' and the user is logged in as 'Usuário: suporte'. The interface is divided into two main sections: 'Árvore | Linear' on the left and 'Síntese' on the right. The 'Árvore | Linear' section shows a tree view with 'Início', 'Laboratório (2013-02-04)', and 'Novo Síntese'. The 'Síntese' section contains a form titled 'Parâmetros de Entrada' with the following fields: 'Massa Inicial:' (unit: g), 'Gás de Arraste:' (unit: g), 'Fluxo de Arraste:', 'Fonte de Carbono:', 'Tipo de Barco:', 'Catalisador:', 'Temperatura de Crescimento:' (unit: °C), 'Tempo de Crescimento:' (unit: s), and 'Fluxo da Fonte de Carbono:' (unit: sccm). Below these fields is an 'Observações:' section with a text input area.

**Figura 5.22.** Oitenta atributos que compõem a atividade de síntese do workflow de LabSem.





## Capítulo 6

# Conclusão e trabalhos futuros

Neste trabalho, mostramos que a nanotecnologia carece de ferramentas computacionais capazes de apoiar o pesquisador em suas atividades. Como resultado, a gestão das informações geradas pelos laboratórios é prejudicada, impedindo um melhor aproveitamento dos dados e dificultando a descoberta de conhecimento. Além disto, existem diversos fatores, tecnológicos e econômicos, que impedem a construção de nanomateriais inovadores. Com o objetivo de suprir estas carências, neste trabalho foi desenvolvido uma ferramenta computacional denominada *NanoTrack*.

O *NanoTrack* é um LIMS, baseado na plataforma Flux®<sup>®</sup>, que foi desenvolvido para o apoio a nanotecnologia. Este sistema computacional é responsável por gerenciar os dados de laboratórios de pesquisa em nanotecnologia, a fim de apoiar a criação de nanomateriais. Esta provê uma plataforma de armazenamento com controle de acesso aos dados que irá melhorar a qualidade e velocidade em que os dados são armazenados e acessados.

Além do gerenciamento das informações do laboratório, também foi apresentado neste trabalho a capacidade do *NanoTrack* em incorporar novas funcionalidades através da utilização de *plugins*. Para mostrar tal capacidade, alguns *plugins* foram desenvolvidos e integrados à ferramenta. Com isto, nos *workflows* dos laboratórios modelados nos estudos de casos deste trabalho, foi possível criar um ambiente de simulação formado por simuladores que foram encapsulados em *plugins*.

Os simuladores desenvolvidos obtiveram resultados bastante satisfatórios, com destaque ao “Simulador Quaternário” que, além de estar presente em um *workflow* modelado para um laboratório real, obteve resultados promissores e com alta precisão.

Além dos simuladores desenvolvidos, também foi criado um algoritmo de busca alternativa. Esta busca é caracterizada por suportar margens de erros e que não é atendida pelos algoritmos convencionais. Através dela, resultados que poderiam satisfazer

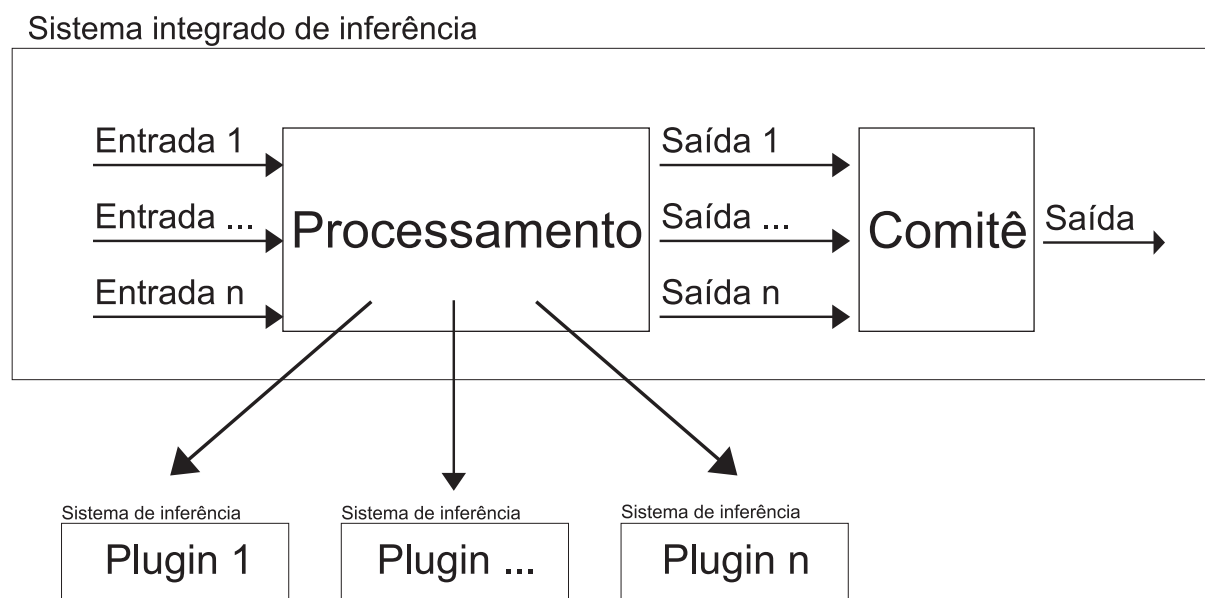
a necessidade do pesquisador e não eram obtidos passam a ser mostrados.

## 6.1 Trabalhos futuros

Com o desenvolvimento deste trabalho, algumas possibilidades foram observadas. Foram realizados três estudos de casos, mas novos laboratórios de nanotecnologia também podem ser modelados a fim de disseminar o uso da ferramenta. Seria muito interessante se num mesmo *workflow* fossem integrados mais de um laboratório. Assim, as atividades seriam realizadas por laboratórios diferentes, mas haveria uma integração entre eles.

Além disto, para que o ambiente de simulação proposto nesta dissertação fique cada vez mais robusto, é importante que sejam desenvolvidos novos simuladores que utilizem técnicas alternativas para descoberta de conhecimento. Estas novas técnicas poderão explorar deficiências que os algoritmos desenvolvidos possam apresentar. Além disto, a diversidade de técnicas poderá possibilitar alternativas para o pesquisador.

Com a implementação de outras técnicas para inferência de dados um “Sistema Integrado de Inferência” pode ser criado. Este seria composto pelos diversos algoritmos encapsulados em um *plugin* e as saídas obtidas após a execução deles seriam passados para um comitê avaliador. Este comitê será responsável por receber os resultados de todos os algoritmos de inferência utilizados e disponibilizar os melhores resultados para o usuário. A saída do comitê pode ser o resultado do sistema de inferência com o menor erro ou uma combinação estatística dos resultados de mais de um sistema de inferência. Veja na Figura 6.1 um exemplo da estrutura do “Sistema Integrado de Inferência” proposto.



**Figura 6.1.** Modelo de um sistema integrado de inferência.



# Referências Bibliográficas

- Asproulis, N. & Drikakis, D. (2009). Nanoscale materials modelling using neural networks. *Journal of Computational and Theoretical Nanoscience*, 6(3):514--518.
- Bakucz, P.; Yacoot, A.; Dziomba, T.; Koenders, L. & Krüger-Sehm, R. (2008). Neural network approximation of tip-abrasion effects in afm imaging. *Measurement Science and Technology*, 19(6):065101.
- Bell, G.; Hey, T. & Szalay, A. (2009). Beyond the data deluge. *Science*, 323(5919):1297--1298.
- Bennett, K. & Rajlich, V. (2000). Software maintenance and evolution: a roadmap. Em *Proceedings of the Conference on the Future of Software Engineering*, pp. 73--87. ACM.
- Botelho, C. (2012). Definição de visões em um lims orientado a fluxo de trabalho para gerenciamento de dados e processos laboratoriais.
- Crnkovic, I.; Larsson, S. & Stafford, J. (2002). Component-based software engineering: building systems from components at 9th ieee conference and workshops on engineering of computer-based systems. *ACM SIGSOFT Software Engineering Notes*, 27(3):47--50.
- Cupertino, L.; Vilela Neto, O.; Pacheco, M.; Vellasco, M. & d'Almeida, J. (2011). Modeling the young modulus of nanocomposites: A neural network approach. Em *Neural Networks (IJCNN), The 2011 International Joint Conference on*, pp. 1599--1605. IEEE.
- Dashtbayazi, M.; Shokuhfar, A. & Simchi, A. (2007). Artificial neural network modeling of mechanical alloying process for synthesizing of metal matrix nanocomposite powders. *Materials Science and Engineering: A*, 466(1):274--283.

- de la Iglesia, D.; Harper, S.; Hoover, M.; Klaessig, F.; Lippell, P.; Maddux, B.; Morse, J.; Nel, A.; Rajan, K.; Reznik-Zellen, R. et al. (2011). Nanoinformatics 2020 roadmap.
- Dias, D.; Pacheco, M. & de Janeiro-RJ-Brasil, R. (2010). Programação genética linear com inspiração quântica.
- Dias, D. M.; Singulani, A. P.; PACHECO, M. A. C.; SOUZA, P. L.; Pires, M. & Vilela Neto, O. P. (2011). Self-assembly quantum dots growth prediction by quantum-inspired linear genetic programming. Em *Evolutionary Computation (CEC), 2011 IEEE Congress on*, pp. 2075--2082. IEEE.
- Guide, M. (1998). The mathworks. *Inc., Natick, MA*, 5.
- Haykin, S. (2001). *Redes neurais-princípios e prática*. Bookman.
- Hey, A. (2010). The fourth paradigm. Em *[plenary] 31st Annual IATUL Conference, West Lafayette*.
- Hey, T. e. a. e. (2009). *The fourth paradigm: data-intensive scientific discovery*. Microsoft Research Redmond, WA.
- Hinton, M. (1995). Laboratory management systems. *New York*.
- Hollingsworth, D. et al. (1995). Workflow management coalition: The workflow reference model. *Document Number TC00-1003*, (1.1).
- Informatics, C. (2013). Core lims.
- Inmetro (2012). *Princípios das boas práticas de laboratório*. BPL.
- Interface, P. (2002). Xml process definition language. *Document Number WFMC-TC-1025 Document Status-XPDL*, 1.
- Kent, S. (1994). Sloan digital sky survey. *Astrophysics and Space Science*, 217(1):27--30.
- Laboratório de Nanomateriais, F. (2013). Ufmg.
- LabSem (2013). Puc-rio.
- Lims, S. (2013). Sql lims.

- Ludäscher, B.; Altintas, I.; Berkley, C.; Higgins, D.; Jaeger, E.; Jones, M.; Lee, E.; Tao, J. & Zhao, Y. (2006). Scientific workflow management and the kepler system. *Concurrency and Computation: Practice and Experience*, 18(10):1039--1065.
- Melo, A.; Faria-Campos, A.; DeLaat, D. M.; Keller, R.; Abreu, V. & Campos, S. (2010). Sigla: an adaptable lims for multiple laboratories. *BMC genomics*, 11(Suppl 5):S8.
- Nateri, A.; Dadvar, S.; Oroumei, A. & Ekrami, E. (2011). Prediction of silver nanoparticles diameter synthesized through the tollens process by using artificial neural networks. *Journal of Computational and Theoretical Nanoscience*, 8(4):713--716.
- of Energy Office of Science, U. D. (2012). Human genome project information.
- Oinn, T.; Greenwood, M.; Addis, M.; Alpdemir, M.; Ferris, J.; Glover, K.; Goble, C.; Goderis, A.; Hull, D.; Marvin, D. et al. (2006). Taverna: lessons in creating a workflow environment for the life sciences. *Concurrency and Computation: Practice and Experience*, 18(10):1067--1100.
- Pitts, W. & McCulloch, W. (1947). How we know universals the perception of auditory and visual forms. *Bulletin of Mathematical Biology*, 9(3):127--147.
- Prog4biz (2013). Bookitlab.
- Rosenblatt, F. (1962). Principles of neurodynamics.
- Singulani, A.; Vilela Neto, O.; Aurélio Pacheco, M.; Vellasco, M.; Pires, M. & Souza, P. (2008). Computational intelligence applied to the growth of quantum dots. *Journal of Crystal Growth*, 310(23):5063--5065.
- Skobelev, D.; Zaytseva, T.; Kozlov, A.; Perepelitsa, V. & Makarova, A. (2011). Laboratory information management systems in the work of the analytic laboratory. *Measurement Techniques*, 53(10):1182--1189.
- Specification, B. (2006). Omg final adopted specification.
- Teamsolutions, T. (2011). Together workflow editor.
- Thorun, N.; Faria-Campos, A.; Alves, V. & Campos, S. V. A. (2013). Fluxar - uma ferramenta dinâmica para a construção de workflows de experimentação biológica. Em *Trabalho de Conclusão de Curso*. PUC - Minas.

- Vilela Neto, O. & Pacheco, M. (2012). *Nanotecnologia Computacional Inteligente: Concebendo a Engenharia em Nanotecnologia*, volume 1. Interciência - Puc Rio.
- Wermelinger, M.; Yu, Y. & Lozano, A. (2008). Design principles in architectural evolution: a case study. Em *Software Maintenance, 2008. ICSM 2008. IEEE International Conference on*, pp. 396--405. IEEE.
- WfMC (2012). Workflow management coalition.
- Zhang, J.; Xu, C.; Yi, M. & Fang, B. (2012). Design of nano-micro-composite ceramic tool and die material with back propagation neural network and genetic algorithm. *Journal of materials engineering and performance*, 21(4):463--470.