2017 International Nuclear Atlantic Conference - INAC 2017
Belo Horizonte, MG, Brazil, October 22-27, 2017
ASSOCIAÇÃO BRASILEIRA DE ENERGIA NUCLEAR – ABEN

Agência Brasileira do ISBN
ISBN 978-85-99141-07-6

CONTROL SYSTEM AND AUTOMATION OF THE SPECTRUM ACQUISITION FOR A PERTURBED GAMMA-GAMMA ANGULAR CORRELATION SPECTROMETER

Ademir F.A. Stachowski, André L. Lapolli, Artur W. Carbonari e Rajendra N. Saxena

Instituto de Pesquisas Energéticas e Nucleares (IPEN / CNEN - SP)
Av. Professor Lineu Prestes 2242
05508-000 São Paulo, SP
alapolli@ipen.br

ABSTRACT

This work reports results from the automation of data acquisition for a gamma-gamma perturbed angular correlation (PAC) spectrometer located in the hyperfine interactions laboratory (LIH) of IPEN. A PAC spectrometer comprises usually four scintillation detectors (BaF_3 for example) and carries out time measurements from gamma rays of any two combinations of detectors in coincidence. Gamma rays come from the decay of probe nuclei embedded into the material to be studied. Measurements are generally performed as a function of temperature. The fourth-detectors PAC spectrometer was automated in Java language because it is a multiplatform and object oriented programming language, having, therefore, portability and longer lifetime due to the ease of implementation of peripherals. Finally, the software has a friendly and intuitive interface with the user being able to operate the hardware manually or automatically with time acquisitions up to 100 consecutive hours. Before each acquisition, the software sets the temperature, waits for temperature stability and records data in different files. Moreover, the software also saves in a file all information of relevant parameters during acquisition.

1. INTRODUCTION

Because the fast development of technology, electronics and computer science, the process of acquisition and analysis of experimental data in all areas of science has been improving in order to obtain increasingly reliable and accurate results. Due to the high degree of integration between the hardware and the software in an increasingly intelligent way based on specific algorithms developed for the data acquisition process, human interference is therefore reduced, avoiding gross errors. On the other hand, computers with increasing memory and processing capacity allow for the analysis of a greater amount of data obtaining results with greater precision.

The present paper reports the development of a data acquisition system implemented to control, collect and analyze data from perturbed angular gamma-gamma correlation (PAC) spectroscopy the study hyperfine interactions in different materials such as intermetallic compounds, nanostructures, macromolecules, biomolecules, etc., in the Laboratory of Hyperfine Interactions (LIH) of IPEN [1-4].

In order to meet the requirements of software development based on computer science, this work uses software engineering, which is a field of computer science focused on the specification, development and maintenance of software systems that apply technologies and practices of computer science, management of projects and other disciplines, aiming at organization, productivity and quality [5].

The acquisition of PAC spectroscopy data requires gamma radiation detectors as well as a specific electronic system. Samples are measured under two conditions: a fixed temperature or in a range of varying temperatures. Two temperature control systems area available in the LIH, one that varies from ambient temperature to 1200 °C and another from ambient down to 8 k. In order to automatize the data acquisition, a computer must control all the processes, setting the temperature, wait temperature stabilization, start acquisition, stop it and save the data and start again. Moreover, all the hardware involved in the processes must be renewed due to the expiration of its useful life and a new acquisition does not always have the same characteristics of data transmission. Therefore, due to this requirement to develop a software with a long lifecycle, aiming to change only small parts of it without the development of a new system for each hardware renovation, we have opted for the object-oriented programming. The language adopted was java that allowed the creation of interfaces and specific methods to control each of the peripherals that involve the system, allowing a constant implementation. In addition, the Java language, due to its characteristics, is a multiplatform language that can be used within several operating systems without the need to rewrite the code.

Finally, the work here reported resulted in a software where it is possible to acquire PAC spectroscopy data automatically or manually, depending on the need. In the next sections, we present the theoretical basis used for the development of the system, the necessary requirements, the design, the results and finally the conclusions.

2. THEORETICAL BASES

Software engineering is the set of techniques, methods, processes and tools used in the specification, construction, implementation and maintenance of software to guarantee the management, control and quality of artifacts generated through human resources [6].

A software is, therefore, an artifact that takes life cycle models into account. Based on a detailed bibliographic search, the incremental iterative model [6] is the one that best meets the requirements of structure and specificity to be implemented in the developed system. In this model, the functionalities are specified according to the rules of cohesion and coupling, respecting the rule of not interacting exhaustively in one functionality before proceeding with another. It is important to emphasize that this model is less vulnerable to changes in specifications throughout development. Figure 1 represents the process of the Incremental Iterative model.

To develop the acquisition system by using software engineering, it is necessary to adopt a programming paradigm, and the paradigm chosen was object orientation, which is a form of software programming adopted by most modern programming languages. It seeks to express things in a way closer to real life, based on the composition and interaction between objects [6].

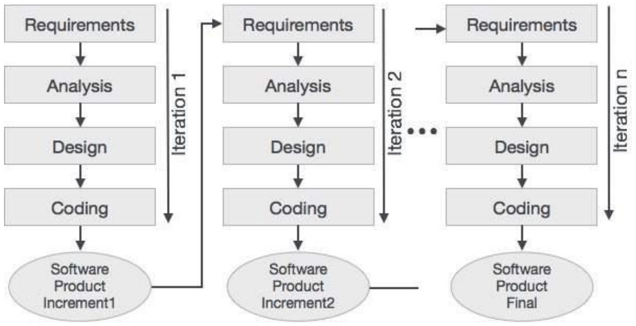

Figure 1: Incremental iterative model.

Object orientation was born as a programming language. The initial basic idea behind object orientation was to unify in modules both the data and the functions that operated the data later this concept was called encapsulation of objects. However, the key concept of an object-oriented language is called polymorphism, which is the principle by which two or more classes derived from the same superclass can invoke methods that have the same identification (signature), but distinct behaviors specific for each derived class using a reference to an object of the superclass type. This mechanism is fundamental in object-oriented programming, allowing to define functionalities that operate generically with objects, avoiding particular details when they are not necessary [5,6]. The benefit of an object-oriented language became clear, but it was not sufficient to meet software engineering requirements. In this context, object-oriented modeling gained momentum. With this approach, the object orientation was boosted and the real benefits prompted by object orientation appeared with increasing ease. However, there was no standard established yet of what an object-oriented modeling would be, and from this need was born the UML (Unified Modeling Language) [6].

Aiming at the ease of communication between different devices in the project as well as the scientific requirements of development, the modeling was performed in UML, which is a diagramming language to specify, visualize and document object oriented software systems assisting in the visualization of drawing and communication between objects [7].

Finally, it must be pointed out that object orientation is an attempt to increase the level of abstraction regarding to the operation of the machine (hardware) [6] as can be seen in Figure 2.

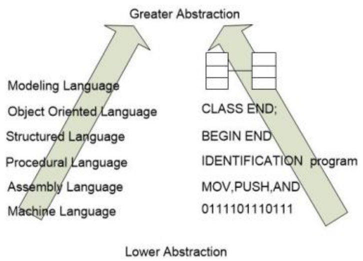

Figure 2: Abstraction levels.

3. REQUIREMENTS TO DEVELOP THE PROJECT

For the development of the project it was necessary to integrate the automation and control of the following devices:

• Acquisition device: multichannel (ORTEC, model 918-A) for PAC spectroscopy data acquisition that communicates via Universal Serial Bus (USB) interface.

• Cryogenic systems: temperature control (Lakeshore, model 331) with range from 8 K to 295 K, which communicates via serial interface RS 232 and RS 485.

• Heating system: temperature controller (Contemp, model CPM45) with temperature range from 20 ° C to 1200 ° C, which communicates via RS 485 serial interface.

These devices were automated and controlled by a computer that uses the Windows XP operating system, where the functionality of these devices are centralized in a single software, as shown in Figure 3.

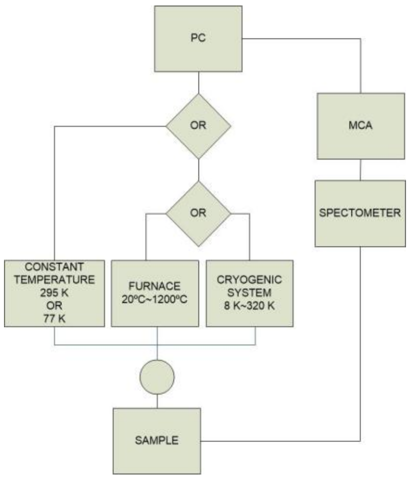

Figure 3: Sketch displaying the devices used for data acquisition.

Under these conditions, the system was developed comprising control modules for the multichannel and the temperature control systems that are detailed in the following section.

4. DEVELOPMENT OF THE PROJECT

The system was designed so that the user or the researcher can access, from a single software, all the necessary functions to carry out PAC measurements and save data in a file for later analysis. Moreover, the user can utilize the manual mode where there is the possibility of set the temperature of the sample via controllers (CPM45 or Lakeshore), starting the measurement when it reaches the specified temperature observing the graphic temperature monitor implemented in the system and later saving the Result of the measurement on file.

In the automatic mode, the user can program up to twenty measurement segments, each one with specific time measurement, temperature with partial data stored every the time period defined by the user. It is worth to emphasize that in both modes (manual and Automatic) it is possible to manually save the results of the measurement in progress manually and to monitor the evolution of the measurement by the graphical representation of the spectrum, which is updated periodically.

The general structure of the system above described is displayed in Figure 4 using the constant-use case diagram of UML, which demonstrates how the different actors will interact with the software [4].

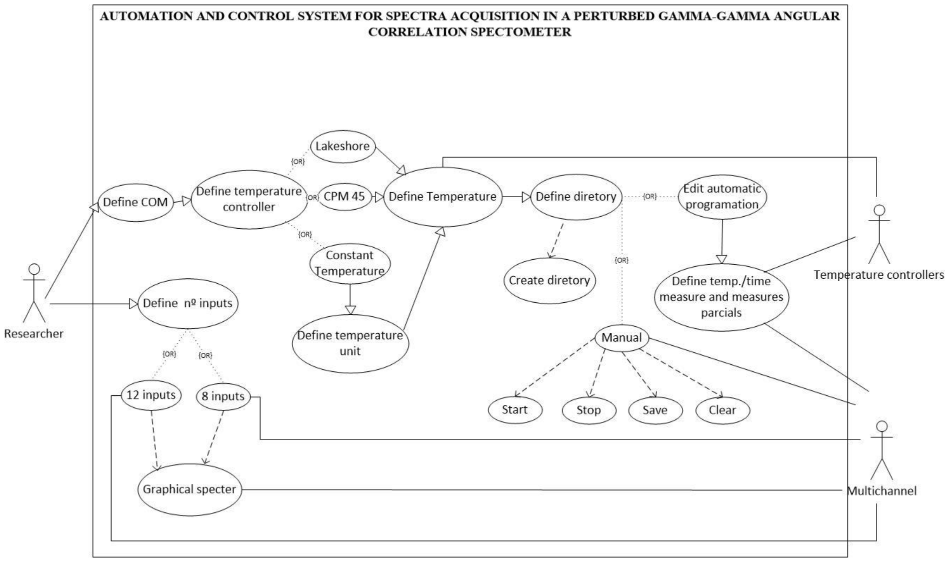

Figure 4: General structure of the project.

4.1. Temperature controllers

The communication interfaces of the temperature controllers are:
• Cryogenic systems: RS 232.
• Heating system: RS 485 (ModBus-RTU protocol).

In our project it was used the RXTX, which is a native library that provides serial and parallel communication (with the possibility of Universal Serial Bus (USB) communication) is used in the Java Development Kit [8].

In addition to setting the desired temperature, it also returns the current value of the process temperature. The Java RXTX API is idealized through the abstract CommPort class and two subclasses, serialPort and ParallelPort, which describes the two types of enabled communication, which are the most common ports on desktops. There is no possibility of instantiating one of these classes directly because their constructors are intentionally not public. To get them, one must use the static method CommPortIdentifier.getPortIdentifier() to capture the available ports, giving the user the opportunity to choose the port to be used. This way, through the CommPortIdentifier, one can call the open() method to receive an instance of the CommPort object. To use the object as a serial port, it is only necessary to cast a cast for any of the sub-classes derived from CommPort. Each of these subclasses has its characteristics and methods that allow the user to use standard Java events to receive notification of available data or even specific types of events as a signal indicator (in data communication) [8-9].

4.2 Multichannel

The interface between the multichannel and the operating system is via Universal Serial Bus (USB) and it is provided by the Component Object Model (COM), which is a Microsoft platform for software components released in 1993. It is used to enable inter-process communication as well as the creation of dynamic objects in any programming language that supports this technology [10].

We used the JACOB library, which is a Java COM bridge that allows calling COM automation components from Java. It uses the Java Native Interface (JNI) to make native calls to the COM and Win32 libraries [11,12].

An auxiliary DLL written in Assembly language has been developed that allows the access to the command sent to the multichannel and obtaining the data returned by it. These commands can be: "START", "STOP", "SAVE" and "CLEAR" the spectrum, emphasizing that the command "SAVE" will record the spectrum in a file at any time of the measurement, both in automatic and manual mode.

Another feature integrated into the system is the possibility of graphically representing the spectrum, which is updated periodically. This functionality was used in the JFreeChart library, which allows the creation of a wide variety of Java language graphics [13].

5. THE SOFTWARE

The main graphical interface can be seen in Figure 5A. From this interface all system functions are accessed, such as: the working directory setting, the directory menu command, temperature controller, number of segments, scroll-down input; if measurement will be in automatic or manual mode, manual and automatic menu commands, among other features. It is important to note that the functions are enabled in a certain order (to prevent accidental activation of some device) namely: first one choose the workbook, number of segments, temperature controller port, scroll-down temperature port. After that, the other commands of the main menu (Manual, Automatic) and the temperature control box (Temperature Controller) that can be used from this moment on, are enabled at any time during the acquisition process.

The Help command displays all the guidelines for using the program, and the Help window, not displayed, starts together with the main window and can be called at any time during program execution.

The use of the temperature control system is exclusive in which, regardless of the form of acquisition, one of the ways of submitting the temperature of the sample is chosen according to the need of the work.

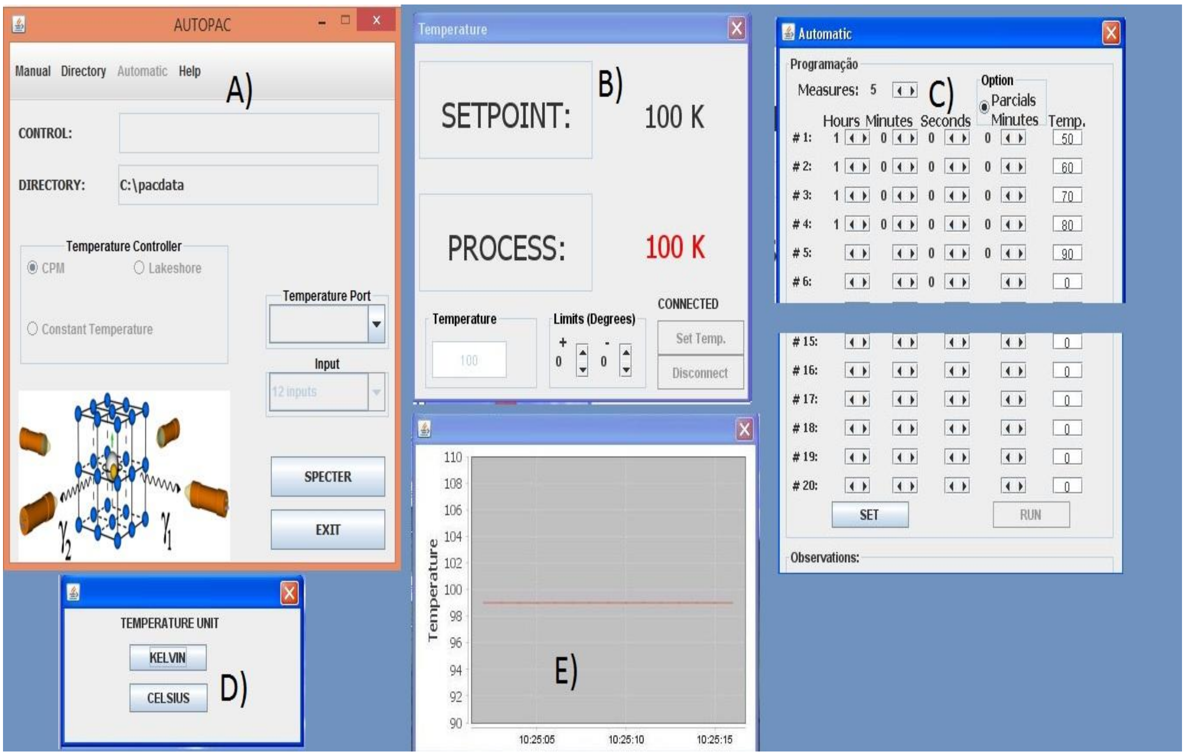

Figure 5: Interfaces of access for users.

The user window (see Figure 5B) is displayed after the determination of the measuring temperature mode. For CPM or Lakeshore controllers, after manual temperature setting, the real-time temperature monitor (Figure 5) appears for the operator to track their variation through the graphics window. When choosing a constant temperature, the operator can choose the temperature in Kelvin or Celsius by pressing one of the buttons of the temperature unit window (Figure 5D).

The option to use the system in manual mode is chosen through the Manual menu command in the main window (Figure 5a). Through this, the user can start or stop the acquisition besides saving the spectrum at any time.

In the automatic mode (Automatic menu command) after choosing the temperature controller, there is the option, in a specific window (see Figure 5C), to program up to 20 steps, each with its respective time measurement and temperature (defined by the user), with the possibility to obtain partial measures (saved in own files) at predefined intervals. It is noteworthy that prior to the beginning of the measurements there is a check so that the temperatures entered are not outside the range accepted by the temperature controllers (20 ° C-1200 ° C at CPM45 and 8 K-320 K at Lakeshore). Each of these measurements are started automatically after stabilization of the set temperature.

By actuating the [SPECTER] button (Figure 5A), the coincidence spectra (Figure 6) can be observed in real time by the user, allowing verification of the measurement progress, thus allowing operator interference at any time during data acquisition.

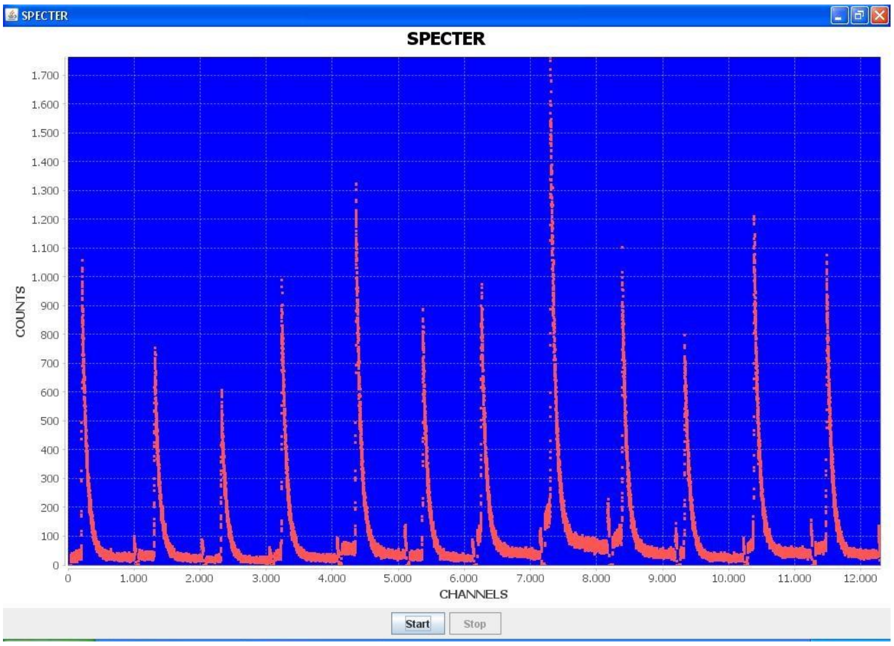

Figure 6: Coincidence spectra created during the PAC spectroscopy data acquisition.

During the acquisition, the program automatically generates a monitoring file where the user can check, during or after the acquisition of data, the defined set of parameters as well as the process of automatic periodic recording of the spectra. If there is any occurrence during the measurement, it is possible, based on this file, to perform a fault diagnosis.

All of the features described above can be seen in the Activity Diagram (in accordance with UML) depicted in Figure 7.

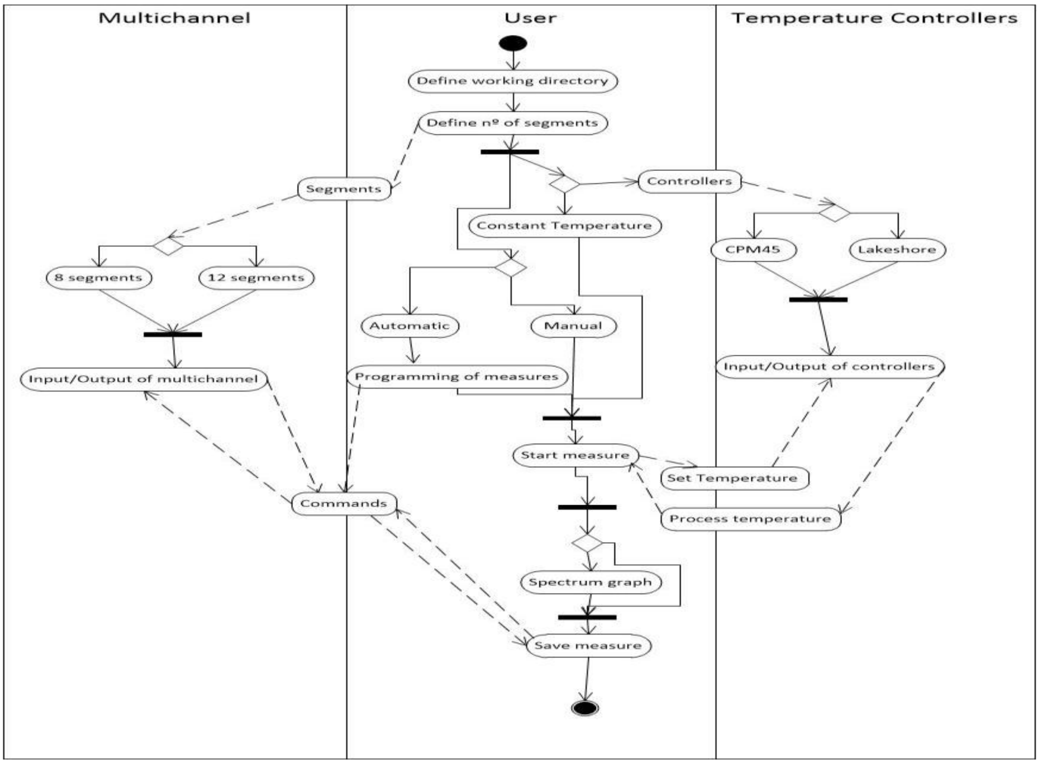

Figure 7: Diagram of activities.

6. CONCLUSIONS

The software was tested in several PAC experiments and was evaluated from the standpoint of software engineering and by the researcher users, and was considered approved.

Due to restrictions imposed by the multichannel manufacturer (ORTEC), the communication is only compatible with several versions of the 32 or 64 Bit Windows platform (XP or higher), but in the future new implementations such restrictions can be eliminated making the multichannel communication totally multiplataform. In the communication module of temperature controllers, this limitation does not exist, being compatible with Linux and Mac platforms, as well as Windows.

REFERENCES

1. Junqueira, Astrogildo de C.. Estudo de interacões hiperfinas em óxidos perovskitas do tipo La(MT)Osub(3) (MT=metais de transicão Fe, Cr, Mn e Co). 2004. Tese (Doutoramento) - Instituto de Pesquisas Energéticas e Nucleares - IPEN/CNEN-SP, São Paulo. 112 p. Orientador: Artur Wilson Carbonari.
2. Attili, R.N.; SAXENA, R.N.; CARBONARI, A.W.; MESTNIK FILHO, J.; UHRMACHER, M.; LIEB, K.P.. Delafossite oxides ABOsub(2)(A=Ag, Cu;B=Al, Cr, Fe, In, Nd, Y) studied by perturbed-angular-correlation spectroscopy using a sup(111)Ag(Beta sup(-)) sup(111)Cd probe. Phys. Rev. B, , v. 58, n. 5, p. 2563-2569, 1998.

3. Cavalcante, Fabio H. de M.. Estudo de interacoes hiperfinas magneticas em sistemas intermetalicos do tipo RAg (R=terra rara). 2004. Dissertacao (Mestrado) - Instituto de Pesquisas Energeticas e Nucleares - IPEN/CNEN-SP, Sao Paulo. 77 p. Orientador: Artur Wilson Carbonari
4. Lapolli, Andre L.. Estudo de interacoes hiperfinas em compostos intermetalicos Gd(nI, Pd)In e Ho(ni, Pd)In. 2006. Tese (Doutoramento) - Instituto de Pesquisas Energeticas e Nucleares - IPEN-CNEN/SP, Sao Paulo. Orientador: Arthur Wilson Carbonari.
5. Pressman, Roger S., *Engenharia de Software*, McGraw-Hill, São Paulo Brasil (2006).
6. Magela, R., *Engenharia de Software Aplicada: Fundamentos*, Alta Books, Rio de Janeiro (2006).
7. Lima, A.S., *UML 2.0 Do Requisito à Solução*, Editora Érica Ltda, São Paulo Brasil (2005).
8. RABELLO, L. M, *Programa em Linguagem JAVA para comunicação Serial*, Embrapa Comunicado Técnico, São Paulo Brasil(2009).
9. "RXTX Wiki", http://rxtx.qbang.org/wiki/index.php/Using_RXTX (2015).
10. "The Component Object Model", https://msdn.microsoft.com/pt-br/library/windows/desktop/ms694363(v=vs.85).aspx (2015).
11. "The JACOB project: A Java-COM bridge", http://danadler.com/jacob/ (2015).
12. "Java-COM integration with JACOB using XML wrappers", http://users.jyu.fi/~minurmin/javacom/javacom_report.pdf (2015).
13. "JFreeChart", http://www.jfree.org/jfreechart/ (2016).