Regras Para a Elaboração do Relatório de Projeto/Estágio da LEEC

Pedro João De-Francesco Resende Fortuna Assis



Departamento de Engenharia Eletrotécnica

Instituto Superior de Engenharia do Porto

2020

Este relatório satisfaz, parcialmente, os requisitos que constam da Ficha de Unidade Curricular de Projeto/Estágio, do 3º ano, da Licenciatura em Engenharia Eletrotécnica e de Computadores

Candidato: Pedro João De-Francesco Resende Fortuna Assis, Nº 1234567, pfa@isep.ipp.pt

Orientação científica: Pedro João De-Francesco Resende Fortuna Assis, pfa@isep.ipp.pt

Empresa: xxxxxx

Orientador: Pedro João De-Francesco Resende Fortuna Assis, pfa@isep.ipp.pt



Departamento de Engenharia Eletrotécnica

Instituto Superior de Engenharia do Porto

4 de julho de 2020

Agradecimentos

Agradecimentos que sejam devidos. Esta secção é opcional!

Resumo

No trabalho efetuado foi proposto que projetasse e implementasse que proporcionasse um melhor controlo das diferentes linhas de produção, pretendia-se que o sistema encurtasse a necessidade de controlo de produção e supervisão, redução do tempo de paragem das linhas, e aumento do controlo de qualidade, com o objetivo de melhorar o rendimento da produção.

O projeto foi dividido em três partes principais, a primeira consistia na conversão do sistema previamente existente para funcionar numa máquina com sistema operativo baseado em Linux.

A segunda parte consistiu na realização de uma interface Web que disponibilizasse, em tempo real, os dados de saída do sistema de controlo de produção para os funcionários e para o supervisor.

A última fase correspondeu á implementação de uma interface gráfica web que permitisse visualizar os dados de forma mais simpática e organizada, que permitisse a sua melhor compreensão por parte dos responsáveis.

Palavras-Chave

Lista, separadas por vírgulas, de palavras, frases, ou acrónimos chave no âmbito do trabalho descrito neste texto.

Abstract

The abstract should summarize the report’s contents, in what concerns the problem identification and/or the formulated hypotheses. The solution, its validation and assessment should also be briefly focused. In all, it should be less than 2 pages in length.

Keywords

We would like to encourage you to list your keywords, key phrases and most relevant acronyms in this section.

Índice

Agradecimentos i

Resumo iii

Abstract v

Índice vii

Índice de Figuras ix

Índice de Tabelas xi

Acrónimos xiii

1. Introdução 1

1.1. Contextualização 3

1.2. Objectivos 4

1.3. Calendarização 4

1.4. Organização do relatório 5

2. Formatos & Estilos 7

2.1. Dimensões da página 8

2.2. Capa 8

2.3. Contracapa 8

2.4. Agradecimentos, resumo e abstract 9

2.5. Índices 9

2.6. Acrónimos 9

2.7. Lemas, proposições e teoremas 10

2.8. Figuras e tabelas 10

2.9. Fórmulas 11

2.10. Extractos de programas 12

2.11. Listas 12

2.12. Notas de rodapé 12

2.13. Referências 13

2.14. Anexos 13

3. Conclusões 15

Referências Documentais 17

Anexo A. Colector baseado em JMS 19

Histórico 21

Índice de Figuras

Figura 1 Arquitetura do OSS 5620 SAM-O [1] 4

Figura 2 Diagrama de estados [7] 11

Índice de Tabelas

Tabela 1 Calendarização do projeto 5

Tabela 2 Descrição dos campos da Notificação de Alarme [7] 11

Acrónimos

|  |  |  |
| --- | --- | --- |
| API | – | Application Programming Interface |
| ASCII | – | American Standard Code for Information Interchange |
| ASN.1 | – | Abstract Syntax Notation - One |
| ASR | – | Alcatel Service Router |
| ATM | – | Asynchronous Transfer Mode |
| CADREDE | – | Sistema de Gestão do Cadastro de Rede da PT |
| COTS | – | Components Of The Shelf |
| CPU | – | Communications Processor Unit |
| CRC | – | Cyclic Redundancy Check |
| CRM | – | Customer Relationship Management |
| CSMA/CD | – | Carrier Sense Multiple Access/Collision Detection |
| ER | – | Elemento de Rede |
| eTOM | – | Enhanced Telecom Operations Model |
| FAB | – | Fulfillment, Assurance & Billing |
| FCAPS | – | Fault, Configuration, Accounting, Performance, Security |
| FCS | – | Frame Check Sequence |
| FIFO | – | First In First Out |

1. Introdução
   1. Contextualização

Este projeto [7] surgiu do desejo de realizar um trabalho no âmbito da gestão de redes baseadas na infraestrutura Simple Network Management Protocol (SNMP) [4][5][8][9], mas que simultaneamente se integrasse na plataforma de alarmes do Sistema de Gestão de Alarmes (SGA) da Portugal Telecom (PT).

Por coincidência, a PT tinha acabado de adquirir (à Alcatel) a nova rede de banda larga Internet Protocol (IP)—Multi Protocol Label Switching (MPLS), e assim surgiu a necessidade de integrar os alarmes gerados por esta infraestrutura (Figura 1) no SGA.



1. Arquitetura do OSS 5620 SAM-O [1]

Dada a pouca experiência, e conhecimento, sobre a aquisição de alarmes para este tipo de tecnologias, foi natural a proposta de incluir a integração, e tratamento, dos alarmes desta rede no âmbito do projeto. Esta inclusão refletiu-se no desenvolvimento de um processo de aquisição dos alarmes gerados e de um programa que formate os relatórios de alarme para o formato (normalizado) da notificação de alarme utilizado no SGA.

* 1. Objetivos

O principal objetivo deste projeto é melhorar a qualidade da supervisão de seis linhas de produção e implementar um sistema web que permita apresentar a informação recolhida durante o processo de produção de forma amigável e organizada ao utilizador. Dada a complexidade do objetivo, sentiu-se a necessidade de o subdividir em múltiplas tarefas mais simples, tais como:

* Estudo do modo de funcionamento e configuração de base de dados MySQL
* A conversão do sistema de recolha de dados para o sistema operativo Linux
* Estudo de algumas ferramentas usadas para desenvolvimento Web (JavaScript, JQuery, Python, …)
* Desenvolvimento de uma interface gráfica para display de dados em tempo real
* Desenvolvimento da aplicação Web para analises estatísticas da informação, e ajuda na tomada de decisões
  1. Calendarização

Para uma melhor gestão do tempo disponível para a realização foi concebida a calendarização da tabela 1, ainda na fase de planeamento. Esta inclui tanto as semanas para aprendizagem de conceitos como para o desenvolvimento efetivo do trabalho. Tal como podemos ver no mês de março foi iniciada a primeira parte do projeto com o estudo da linguagem c++, assim como do sistema já implementado na Indmei, e gradualmente foi feita a conversão para Linux.

O mês de abril deu início á segunda fase o projeto, que corresponde ao estudo de algumas linguagens de programação tal como, python, html, css, e JavaScript. Neste mês foi também desenvolvida a interface gráfica destinada às linhas de produção e supervisores.

A terceira fase do trabalho começou no início do mês de maio e prolongou-se até ao final da terceira semana de junho. Esta correspondeu ao desenvolvimento da aplicação web destinada á apresentação e analise estatística dos dados de produção.

**Tabela 1 Calendarização do projeto**

Uma imagem com computador, branco, armário, portátil

Descrição gerada automaticamente

* 1. Organização do relatório

No primeiro capítulo é feita uma breve introdução ao projeto realizado no âmbito da unidade curricular de Projeto/Estágio do 3º ano da licenciatura em engenharia eletrotécnica e de computadores (LEEC), assim como a contextualização do problema a resolver, calendarização, e apresentados os principais objetivos propostos.

No capítulo seguinte, 2, analisa-se a arquitetura do sistema, onde são apresentadas as diferentes tecnologias utilizadas e são explicadas as interações entre estas. É também, feito o enquadramento destas no sistema em geral.

No capítulo 3, são expostos os principais algoritmos, etapas, e processos utilizados no desenvolvimento do projeto, assim como uma descrição detalhada dos mesmos.

No capítulo seguinte, 4, procede-se á mostra e analise dos resultados obtidos no trabalho.

Por fim, no último capítulo, 5, são retiradas as principais conclusões e apresentadas perspetivas que poderão ser abordadas para desenvolvimentos futuros.

1. Arquitetura do sistema

Neste capítulo é feita a apresentação e descrição das tecnologias utilizadas no desenvolvimento do projeto, assim como a explicação para a escolha das mesmas. Todas as tecnologias utilizadas são gratuitas (open source) mas apresentam muita robustez e, algumas, elevado nível de complexidade. Este aspeto permite reduzir os custos de desenvolvimento do projeto mantendo a sua qualidade.

A componente de recolha de dados já estava implementada e precisou apenas de uma conversão para o seu correto funcionamento em sistema operativo Linux. Esta recolha é feita através de um Arduíno que está constantemente a escrever para a porta serie do PC. Posteriormente o processamento e validação dos dados é realizado por um aplicativo escrito em c++.

Na figura seguinte está representado o diagrama do sistema implementado, onde podemos ver as diferentes tecnologias utilizadas e algumas das suas interações, este foi pensado ainda na fase de planeamento e mais tarde corroborado pela fase de analise dos resultados.

Uma imagem com captura de ecrã

Descrição gerada automaticamente

**Figura 2 Esquema geral do sistema**

Tal como podemos verificar na figura 2, a base de dados MySQL encontrasse no centro do diagrama, tornando evidente a sua importância no sistema. Numa primeira fase os dados recolhidos pelo Arduíno são enviados para a porta serie do PC e posteriormente são processados e validados pelo aplicativo, que é responsável por inserir as entradas de informação na base de dados. Posteriormente a aplicação web comunica com o MySQL para obter os dados relativos á produção, esta comunicação realizasse através de uma application programming interface (API) do Mysql para a linguagem Python.

* 1. MySQL

O MySQL é um sistema de gestão de base de dados relacional (RDBMS) que adota maioritariamente um modelo de cliente-servidor, em que o cliente gera os pedidos e o servidor processa esses pedidos e constrói as respetivas respostas, e a Structured Query Language (SQL). Os comandos SQL dividem-se quatro grandes grupos, Data Definition Language (DDL), Data Manipulation Language (DML), Data Control Language (DCL), e Transaction Control Language (TCL). Neste projeto os grupos mais utilizados são o DDL, uma vez que esta lida com a estrutura e descrição dos dados dentro da base de dados, e o DML que, faz a manipulação dos dados, este inclui os comandos mais conhecidos, tais como SELECT, INSERT, UPDATE, etc..

* 1. Python

Python é uma das linguagens de programação mais interessantes e usadas a nível mundial. De acordo com as tabelas de classificação da TIOBE, python é a terceira linguagem mais popular, ficando apenas atrás de c e java. As principais vantagens desta linguagem em comparação a outras incluem:

* É uma linguagem interpretada, o que significa que não precisa de ser compilada para instruções máquina de baixo nível
* O Python Package Index (PyPI) contêm vários módulos de terceiros que a tornam capaz de interagir com a maioria das outras linguagens e plataformas
* Extensas librarias de suporte: oferece uma grande biblioteca padrão que inclui áreas como protocolos de internet, operações de string, ferramentas de desenvolvimento Web e interfaces de sistema operacional. Muitas tarefas de alto uso já foram escritas na biblioteca padrão o que reduz significativamente o comprimento do código a ser escrito.

[https://www.invensis.net/blog/it/benefits-of-python-over-other-programming-languages/]

Outra vantagem do Python é a sua grande flexibilidade de programação Web, graças a todas as estruturas (frameworks) que tornam o desenvolvimento Web rápido e fácil, como por exemplo Django e Flask. Uma framework é um pacote que contém pacotes e módulos que fornecem cetras funcionalidades que podem ser alteradas de forma seletiva para criar um software específico de aplicação.

1. **Flask**

Uma Web Application Framework é uma coleção de módulos e librarias que possibilitam aos desenvolvedores escrever aplicações com maior nível de abstração em relação a detalhes de baixo nível, tal como protocolos e gestão de threads. [https://pythonbasics.org/what-is-flask-python/]

Flask é uma Web Framework desenvolvida por Armin Roonacher, que lidera uma equipa internacional de entusiastas Python chamados Poocco. Esta é baseada no Wekzeg web server gateway interface (WSGI) kit e no mecanismo de modelo Jinja2. [https://pythonbasics.org/what-is-flask-python/]

![Uma imagem com captura de ecrã, pássaro

Descrição gerada automaticamente]()

**Figura 3 Modelo Jinja2**

O Jinja2 é um mecanismo de modelo para Python usado para criar HyperText Markup Language (HTML), eXtensible Markup Language (XML), ou outros formatos de marcação, que são retornados ao usuário por meio de uma resposta HyperText Transfer Protocol (HTTP). Um modelo contém variáveis que são substituídas pelos valores passados quando o modelo é renderizado, tal como mostra a figura 3. As variáveis são uteis com dados dinâmicos.

Para que a FlaskApp funcione de forma correta é preciso que os ficheiros sigam a organização apresentada na figura 4.

![Uma imagem com mesa

Descrição gerada automaticamente]()

**Figura 4 Estrutura FlaskApp**

A estrutura da aplicação ajuda a manter o código organizado e acessível, por defeito o flask espera que o diretório “templates” esteja na mesma pasta onde o módulo foi criado, no entanto pode ser especificado uma localização diferente através do comando mostrado abaixo.

app = Flask(\_\_name\_\_, template\_folder='../pages/templates')

[https://codeburst.io/jinja-2-explained-in-5-minutes-88548486834e]

* 1. JavaScript

JavaScript é uma linguagem de script que roda do lado do cliente, ou seja, ao invés de rodar remotamente em servidores na internet, o JavaScript tem como característica principal correr os programas localmente, sendo estes executados pelo navegador web. Assim sendo, o JavaScript fornece às páginas web a possibilidade de programação, transformação e processamento de dados enviados e recebidos, interagindo com a marcação e exibição de conteúdo da linguagem HMTL e com a estilização desse conteúdo proporcionada pelo CSS nessas páginas, o que torna a experiência do utilizador mais interessante e interativa.

JavaScript é uma linguagem de thread única, o que significa que possui uma única Call Stack e, portanto, apenas tem a capacidade de fazer uma coisa de cada vez.

O facto desta linguagem receber e enviar pedidos através da rede para servidores remotos, sem ter a necessidade de recarregar a totalidade da página conforme as entradas do usuário foi um elemento decisivo quanto á sua escolha para este projeto.

[<https://www.educba.com/how-javascript-works/>]

1. **JQuery**

JQuery é uma biblioteca JavaScript rápida, pequena e rica em recursos que simplifica o processo de passagem e manipulação de documentos HTML, eventos, animação e AJAX. Tem uma API fácil de usar que funciona em quase todos os navegadores. [<https://jquery.com/>]

O principal motivo da utilização desta tecnologia no projeto é a simplicidade de execução dos pedidos AJAX (asynchronous JavaScript and XML). Estes pedidos possuem grande importância porque permitem a troca de dados enter o front-end e o back-end sem usar alguma variedade de linguagem no lado do servidor para injetar os dados necessário no JavaScript e sem ter de navegar para uma página nova sempre que for preciso atualizar o conteúdo. [<https://www.quora.com/Why-is-JSON-and-AJAX-important-for-JavaScript?share=1>]

$.ajax({

url: "/getconf",

success: function (result) {

console.log(result.postos)

n\_postos = result.postos

n\_turnos = result.turnos

$('#n\_postos').append(n\_postos)

$('#n\_turnos').append(n\_turnos)

}

})

* 1. Agradecimentos, resumo e abstract

Estas secções não são numeradas (mas as respetivas folhas sim, usando para tal numeração romana) pelo que deve usar o estilo “Secção não numerada”. O estilo “Texto” deve ser utilizado para formatar o conteúdo de cada uma destas secções.

Estas secções correspondem a novas secções do documento Microsoft Word™ e por isso devem iniciar-se numa página ímpar (impressão frente e verso). De uma forma geral, este reparo é válido para a totalidade das secções, numeradas e não numeradas, ou seja: capítulos, índices, acrónimos, referências documentais e anexos.

Na secção “Resumo” e “Abstract” existe a possibilidade de indicar um conjunto de termos ou frases que reflitam aspetos considerados relevantes para o trabalho. As subsecções “Palavras-Chave” e “Keywords” não são numeradas e fazem ambas uso do estilo “Keywords”.

* 1. Índices

Devem existir índices correspondentes aos assuntos apresentadas, figuras e tabelas (outros são possíveis[[1]](#footnote-1)). O índice geral é relativo à totalidade das secções numeradas e não numeradas, subsecções e sub-subsecções. O índice de figuras é só relativo ao estilo “Legenda Figura”, enquanto que o índice de tabelas é só relativo ao estilo “Legenda Tabela”.

Para atualizar estes índices basta selecionar, com o botão do lado direito do apontador, o índice desejado e indicar que pretende atualizar este campo. Posteriormente deve escolher uma de duas opções: atualizar o número da página de cada entrada ou então as entradas propriamente ditas.

* 1. Acrónimos

A lista de acrónimos, ou glossário, encontra-se formatada numa tabela de 3 colunas, sendo que o estilo “Texto” se aplica a cada célula da tabela. Para incluir um novo acrónimo deve adicionar uma nova linha à tabela. A tabela de acrónimos não deve apresentar qualquer grelha visível que possa posteriormente aparecer na impressão.

* 1. Lemas, proposições e teoremas

Lemas, preposições e teoremas, devem ser numerados por ordem crescente, iniciando em 1. Deve usar os estilos “Lema”, “Preposição” e “Teorema”.

1. Isto é um exemplo da aplicação do estilo “Lema”. Este estilo é numerado por ordem crescente, iniciando em 1.
2. Isto é um exemplo da aplicação do estilo “Preposição”. Este estilo é numerado por ordem crescente, iniciando em 1.
3. Isto é um exemplo da aplicação do estilo “Teorema”. Este estilo é numerado por ordem crescente, iniciando em 1.
   1. Figuras e tabelas

As figuras devem ser formatadas usando o estilo “Figura”, enquanto que as tabelas devem respeitar o formato descrito no estilo “Tabela”. As legendas das figuras e tabelas devem ser numeradas por ordem crescente, iniciando em 1. Para tal deve usar os estilos “Legenda Figura” e “Legenda tabela”. Note que as legendas das tabelas (ver Tabela 2) antecedem as próprias tabelas, ao contrário, as legendas das figuras sucedem-se às mesmas.

Deve verificar se as linhas utilizadas nas figuras (ver Figura 2), nomeadamente as que são interrompidas, são visíveis depois de impressas. No caso de desejar imprimir o relatório a preto e branco deve optar por figuras em tons de cinzento e com uma resolução adequada à impressora que irá usar. Como regra geral deve usar figuras com uma resolução superior a 800 dpi (de preferência 1200 dpi) [3].



1. Diagrama de estados [7]
2. Descrição dos campos da Notificação de Alarme [7]

|  |  |
| --- | --- |
| Data | Data do Relatório de Alarme no formato: AAAA-MM-DD; |
| Hora | Hora do Relatório de Alarme no formato: HH:MM:SS; |
| Sist\_superv | Tipo de Subsistema de Supervisão (ASR, GES, EWS, S12, …); |
| Gravidade | Tipo ou nível de gravidade do alarme: Crítico (CRI), Maior (MAI), Menor (MEN), Aviso (AVI), Indeterminado (IND) e normal (NOR). Dado que as notificações são em certos casos usadas para a comunicação de outros eventos é de considerar mais um atributo possível: Informativo (INF); |
| Tipo\_notif | Tipo de notificação (I - início do alarme, F - fim do alarme); |

* 1. Fórmulas

As fórmulas encontram-se formatadas em tabelas de 2 colunas, sendo que a 1ª coluna se destina à expressão em si e a 2ª coluna à identificação da mesma. Esta identificação permitirá referenciar qualquer expressão matemática incluída no texto. Assim, as fórmulas devem ser numeradas por ordem crescente, iniciando em 1. O estilo “Equação” deve ser utilizado. A tabela não deve apresentar qualquer grelha visível que possa posteriormente aparecer no documento impresso.

|  |  |
| --- | --- |
| x + y = z . | (**1**) |

Deve pontuar qualquer expressão matemática tal como de um texto se tratasse. Para tal deve deixar um espaço em branco antes de aplicar a pontuação desejada [3].

* 1. Extratos de programas

Sempre que tal se justifique, como forma de melhorar a clareza da exposição, devem ser incluídos, ao longo do texto, extratos, de reduzida dimensão, de programas (código). Neste caso, deve-se usar o tipo de letra “Courier” ou “Courier New”, tamanho 11. Pode utilizar o estilo “Código” para formatar os extratos de programas a inserir.

Chama-se à atenção para evitar a utilização abusiva de extratos de código pois é reveladora da falta de capacidade de seleção e de síntese por parte do autor do texto. Nestas situações é usual o leitor não atribuir o devido valor às “listagens” apresentadas. Sempre que possível deve colocar os programas desenvolvidos num suporte de armazenamento digital (por exemplo, num CD) e entregar este junto com o relatório. Este procedimento permitirá agilizar o processo de avaliação e reduzir a dimensão do relatório.

<start Kprogram>

<data>

<hora>

<sist\_superv>

<gravidade>

<tipo\_notif>

<local>

<fam\_tec>

<instancia>,<componente>,<tipo\_alarme>

<end>

* 1. Listas

As listas (*bullets*) de itens fazem uso do estilo “Lista Bullet”.

* 1. Notas de rodapé

As notas de rodapé devem ser numeradas e a respetiva referência anexa à palavra pretendida (no texto principal). A identificação da referência deve aparecer acima da linha após um eventual sinal de pontuação. Deve usar o tipo de letra “Times New Roman”, tamanho 10 para a nota de rodapé e referência (estilo “Nota de Rodapé”).

* 1. Referências

As referências documentais (estilo “BibItemNum”) são numeradas, fazem uso do tipo de letra “Times New Roman”, tamanho 12 e justificadas à esquerda. Deve organizar as referências documentais por ordem alfabética.

As referências documentais devem incluir (consultar as referências documentais deste texto) os elementos necessários à correta identificação da obra em causa, de forma a permitir a sua consulta. Usualmente as referências são compostas pelos seguintes elementos.

* Autoria
* Título
* Edição
* Local de edição
* Editor
* Data

Deve consultar a Norma Portuguesa (NP) 405, *Informação e documentação: referências bibliográficas*, para mais informação sobre as convenções a aplicar a referências a monografias, partes ou capítulos de monografias (livros), dissertações (licenciatura, mestrado e doutoramento), dicionário e enciclopédias (palavras ou artigos), publicações periódicas, artigos em jornais diários ou semanários, etc.

* 1. Anexos

Devem ser colocados em anexo a listagem detalhada dos resultados compilados, gráficos, listagens de programas, e outra informação complementar que não é da responsabilidade do autor (por exemplo, características dos equipamentos utilizados). Este elemento organizacional do documento é distinto, na sua natureza, dos apêndices. Estes últimos consideram-se um prolongamento do texto principal, cujo conteúdo representa informação adicional sobre um determinado tópico.

Para formatar o título do anexo deve usar o estilo “Anexo”. O conteúdo do anexo faz uso dos estilos definidos para o corpo principal.

Ao longo deste texto foram sendo apresentadas conclusões que permitiram sustentar as opções de desenvolvimento efetuadas ao longo do projeto. Assim, nesta última secção é realizada uma síntese das principais conclusões, consequências e relevância do trabalho realizado e perspetivados futuros desenvolvimentos.

Sob pena de repetição, é inevitável voltar a recordar a condicionante que acabou por determinar a metodologia de desenvolvimento adotada. O constrangimento encontrado no desenvolvimento da solução de recolha de Alarmes através da interface Java Messaging Service (JMS), além de ter resultado num atraso inesperado no desenvolvimento do projeto, impediu explorar as capacidades de correlação dos alarmes, que se propagam por diversos Elementos de Rede (ER), e a determinação da *root-cause* (que assume um papel da máxima importância).

A tradução das *traps* pelo módulo snmptrapd revelou-se de grande utilidade, pois contribuiu para o aumento da produtividade da equipa de desenvolvimento e tem permitido o desenvolvimento de outros coletores de uma forma mais simples e célere.

Em termos funcionais as aplicações desenvolvidas foram verificadas face aos objetivos pretendidos e validadas pelos futuros utilizadores. No que respeita à avaliação do desempenho, a fase de teste correu dentro do esperado, com a manipulação de uma média (próxima) de 16 000 alarmes por dia, que geram uma média de 5 000 notificações por dia.

Referências Documentais

1. Alcatel—*SAM-O Service Aware Manager Open Interface, Release 3.0, OSS Interface Developer Guide*. Alcatel, 2005.
2. AZEVEDO, Carlos; AZEVEDO, Ana—*Metodologia Científica: contributos práticos para a elaboração de trabalhos académicos*. 3ª ed. Porto: Ed. C. Azevedo, 1996. pp 67-91
3. Hofmann, Alfred; Barth, Ursula; Beyer, Ingrid; Günther, Christine; Holzwarth, Frank; Kramer, Anna; Siebert-Cole, Erika—*Lecture Notes in Computer Science: Authors’ Instructions for the Preparation of Camera-Ready Contributions to LNCS/LNAI/LNBI Proceedings*. Springer-Verlag, Computer Science Editorial, Tiergartenstr. 17, 69121 Heidelberg, Germany
4. Mauro, Douglas; Schmidt, Kevin—*Essential SNMP, Help for System and Network Administrators*. O’Reilly, 2001. ISBN 0-596-00020-0.
5. Net-SNMP, Version: 5.2.1, <http://www.net-snmp.org/>
6. PINTO, J. A. Cardoso—*RETA, Manual de Referência*. PT-Comunicações, 2004.
7. RODRIGUES, José—*gestAlarm: Um Colector de Alarmes Para a Rede IP/MPLS da PT*. Dissertação de Licenciatura em Engenharia Eletrotécnica—Eletrónica e Computadores orientada pelo Eng.º Pedro Assis e apresentada no Instituto Superior de Engenharia do Instituto Politécnico do Porto, na especialidade de Telecomunicações em 2006.
8. ROSE, Marshall T. —*The Simple Book: An Introduction to Networking Management, Readings in Simple Network Management Protocol*. Prentice Hall, Inc, 1996. ISBN 0-13-451659-1.
9. Schönwälder, Jürgen—*Internet Management Technologies*. International University Bremen, Germany, <http://www.faculty.iu-bremen.de/schoenw>
10. Coletor baseado em JMS

Neste anexo são descritos alguns dos pormenores do desenvolvimento e testes efetuados para o coletor de alarmes pela interface JMS [1] e respetivos resultados.

Histórico

* 13 de Dezembro de 2006, Versão 1.0, <mailto:pfa@isep.ipp.pt>
* 21 de Junho de 2007, Versão 1.0.a, <mailto:pfa@isep.ipp.pt>

$Id:PESTA\_LEEC\_RELATORIO.dot v1.0.a Date:21-06-2007$

1. Existe a possibilidade de incluir, no final do texto, um índice remissivo. [↑](#footnote-ref-1)