Gestão de projectos TI e administração centralizada de sistemas e redes

Cenários práticos em contexto empresarial

Mário João Gonçalves Antunes

Relatório submetido ao Instituto Politécnico de Leiria para obtenção do título de Especialista na área de Ciências Informáticas (Código 481 - Portaria nº 256/2005 de 16 de Março), de acordo com o DL nº 206/2009 de 31 de Agosto e o Despacho nº8590/2010

E-mail mario.antunes@ipleiria.pt

Morada Institucional Escola Superior de Tecnologia e Gestão
Morro do Lena – Alto do Vieiro
Apartado 4163
2411-901 Leiria

1 de Setembro de 2010

“ Pelo sonho é que vamos,

Comovidos e mudos.

Chegamos? Não chegamos?”

“O Sonho”, Sebastião da Gama.

AGRADECIMENTOS

Ao longo da minha actividade profissional tive oportunidade de conhecer inúmeras pessoas, com quem pude partilhar experiências e consolidar os meus conhecimentos no domínio da informática em geral e da administração de sistemas e redes em particular.

Agradeço às direcções das empresas do grupo Sonae onde tive oportunidade de desempenhar a minha actividade, pela confiança que depositaram em mim e no meu trabalho. Agradeço igualmente a todos os profissionais dos departamentos de gestão de redes e de projectos, com quem tive oportunidade de trabalhar, pelo companheirismo, pela troca de experiências e conhecimentos técnicos, bem como pelo apoio que sempre demonstraram.

Aos Institutos Politécnicos de Leiria e Coimbra, bem como aos Departamentos de Engenharia Informática da ESTG-Leiria e do ISEC, pela forma como me acolheram e pelos projectos e actividades que me deram oportunidade de participar.

Aos meus amigos, aqueles que resistem aos tempos, pelo apoio e incentivo incondicionais.

Aos alunos com quem pude partilhar conceitos, histórias e experiências relacionadas com esta temática.

Por fim, e não menos importante, à minha família, à minha esposa e às minhas filhas, pela ajuda, apoio e carinho.

RESUMO

O crescimento exponencial do número de redes e computadores interligados, verificado a partir de meados da década de 80, bem como a globalização dos mercados e a massificação e popularização da Internet, registada nas décadas seguintes, tornou as organizações altamente dependentes de tecnologias de informação e comunicação. Esta dependência tornou-as reféns de uma infra-estrutura de rede, servidores e serviços, que se pretendia segura, fiável e com mecanismos eficientes de mitigação das falhas. Este facto obrigou as organizações, em especial as empresas, a investirem continuamente em tecnologias de *hardware* e *software* que respondessem às necessidades do negócio, bem como a contratarem profissionais na área de administração e gestão de sistemas, servidores e redes. Nesta área as competências necessárias são vastas e abarcam conhecimentos multi-disciplinares. Genericamente, pretende-se que estes profissionais assegurem duas missões principais. Por um lado, o diagnóstico e a resolução de problemas. Por outro, a implementação de soluções técnicas avançadas que dotem a infra-estrutura de características distintivas de desempenho, segurança e disponibilidade, entre outros aspectos.

Este relatório descreve o meu percurso profissional realizado em contexto empresarial, no período compreendido entre 1988 e 2001, enquanto quadro de várias empresas do grupo Sonae. Durante esse período desempenhei as funções de administrador e gestor de sistemas e redes, nomeadamente de tecnologias proprietárias da IBM (S/36 e S/38), de VAX-VMS e em sistemas Unix (IBM AIX, HP-UX e Linux). Fui igualmente gestor de projectos na área de Tecnologias de Informação em negócios críticos do grupo Sonae. Nesse sentido, tive a meu cargo a coordenação de vários projectos de implementação e gestão de redes de média e grande dimensão, bem como a implementação de soluções de alta disponibilidade em negócios críticos da Sonae Distribuição, recorrendo a tecnologias de *clustering* de servidores Unix.

O documento inicia com uma descrição sumária do contexto empresarial em que se desenrolou a actividade, detalhando-se as tecnologias envolvidas e as topologias de rede geridas. Posteriormente, sintetizam-se os principais conceitos associados à temática da alta disponibilidade em redes, nomeadamente a identificação dos pontos críticos de falha e as várias formas de os mitigar. Concretamente no âmbito da implementação de clusters de alta disponibilidade, são identificados os principais conceitos associados e a metodologia para a sua construção. De seguida são detalhados três projectos principais onde tive responsabilidades de coordenação e gestão: implementação de um tradutor de EDI; implementação da nova plataforma operacional da gestão de entrepostos da Sonae Distribuição; implementação de uma solução centralizada de *backups*. Em cada um deles são enquadrados os principais conceitos técnicos e detalhada a solução implementada. De seguida tecem-se algumas considerações gerais à actividade desenvolvida no período em análise e ao estado actual dos projectos descritos. Por fim, descreve-se a integração da actividade desenvolvida em contexto empresarial com o período posterior a 2001, onde tenho desempenhado as funções de docente no ensino superior politécnico público.

ABSTRACT

The exponential growth of interconnected networks and computers, recorded from the mid-80s and the globalization of markets and mass distribution and popularization of the Internet, verified in following decades, created a dependency of organizations (like companies) on information technology and communications. This dependence turns the companies and individuals indirectly, highly dependent of an infrastructure network, servers and services, which aims to be secure, reliable and to implement efficient mechanisms to mitigate the failures. This has forced companies to invest continuously, not only in hardware and software technologies that meet the business needs, but also on the employment and training of systems and network administrators, able to manage the overall applicational and network infrastructure. In this area of knowledge the skills involved are vast and necessarily involve a multi-disciplinary approach, aiming to ensure two main tasks: the diagnosis and troubleshooting; the implementation of advanced technical solutions to endow the infrastructure with distinctive features of performance, security and availability, among others.

This report describes my career done in the business environment, in the period between 1988 and 2001, as system administrator and senior project leader in various companies of the group Sonae. During this period I served as systems and networks manager of several technologies, including IBM (S/36 and S/38), VAX-VMS and Unix (IBM AIX, HP-UX and Linux). I was also a senior ICT project manager, namely on the implementation of high availability solutions for mission-critical business on Sonae Distribuição, using clustering technologies of Unix servers.

The document begins with a brief description of the business environment in which the activity took place, detailing the managed network topologies. Later, it summarises the main concepts associated with the subject of redundancy, high availability and disaster recovery, including the identification of critical points of failure and the various ways to mitigate them. Particularly on the implementation of highly available Unix clusters, the major concepts and methodology for its deployment are identified. Next, three main projects in which I had management and coordination responsibilities are described: EDI translator; the implementation of an operating platform for managing retail warehouses in Sonae Distribuição; and the implementation of a centralized solution for backups. In each of them the main technical concepts are framed and the technical solution implemented is detailed. The report ends with some conclusions on the activity of the period, emphasizing some general considerations about the current status of the projects and the further integration of the business activity within the academic activity in which I am involved since 2001.

ÍNDICE

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ACRÓNIMOS

ADSM	ADSTAR Distributed Storage Manager
AIX	Advanced Interactive Executive
ARP	Address Resolution Protocol
BCRS	Business Continuous Recovery Service
BDN	Base de Dados de Negócio
CARP	Common Address Redundancy Protocol
COTS	Components Off-The-Shelf
DEC	Digital Equipment Corporation
DNS	Domain Name Service
DRP	Disaster Recovery Plan
EAN	European Article Numbering
ECR	Efficient Consumer Response
EDI	Electronic Data Interchange
EIGRP	Enhanced Interior Gateway Routing Protocol
FCCN	Fundação para a Computação Científica Nacional
GE	Gestão de Entreposto
GELO	Gestão de Loja
GI	Gestão de Inventário
GLBP	Global Load Balancing Protocol
GPFS	General Parallel File System
GPL	GNU Public License
GTI	Gestor de Transacções Integrado
HACMP	High Availability Cluster Multi-Processing
HP	Hewlett-Packard
HPS	High Performance Switch

HSRP	Hot-Standby Redundancy Protocol
IBM	International Business Machines
IETF	Internet Engineering Task Force
LAN	Local Area Network
LVM	Logical Volume Manager
LVS	Linux Virtual Server
MCH	Modelo Continente Hipermercados
MOSEL	Movimento Open Source ESTG-Leiria
MPP	Massive Parallel Processing
NF	Número de Falhas
NFS	Network File System
NLBS	Network Load Balancing Services
NPO	Nova Plataforma Operacional
OSPF	Open Shortest Path First
OSS	Open Source Software
PBL	Pick-By-Line
PSSP	Parallel System Support Programs
QA	Quality Assurance
RAID	Redundant Array of Inexpensive Drives
RF	Rádio Frequência
RFC	Request For Comments
RIP	Router Information Protocol
SCSI	Small Computer Systems Interface
SI	Sistemas de Informação
SIAF	Sociedade de Iniciativa e Aproveitamentos Florestais
SIR	Sonae Indústria e Revestimentos
SLA	Service Level Agreement
SMP	Symmetric Multi-Processor

SNA	System Network Architecture
SOPAS	Sistema de Operações Assistido
SP	Scalable Power
SPDF	Sonae Produtos e Derivados Florestais
SRD	Sonae Redes de Dados
SSA	Serial Storage Architecture
TEDI	Tradutor de EDI
TI	Tecnologias de Informação
TIC	Tecnologias de Informação e Comunicação
TMR	Tempo Médio de Reparação
TSM	Tivoli Storage Manager
TTI	Tempo Total de Indisponibilidade
TTO	Tempo Total de Operação
UPS	Uninterruptible Power Supply
VAN	Value-Added Network
VMS	Virtual Management System
VRRP	Virtual Router Redundancy Protocol
VTL	Virtual Tape Library
WAN	Wide Area Network

1 INTRODUÇÃO

No início da década de 90, o uso massivo de aplicações informáticas no seio das organizações levou a um crescimento exponencial do número de computadores e das redes empresariais de suporte ao negócio. Genericamente, todos os tipos de organizações foram tocados pelas tecnologias de informação e comunicação, desde as empresas de serviços, passando pelas transformadoras, até às produtoras. Esta mudança não foi instantânea, mas antes gradual, desenrolando-se ao longo do tempo. Também a intensidade não foi a mesma, dependendo do tipo de empresa e do modelo de gestão implementado. O aparecimento da Internet e a sua popularização, quer para uso pessoal quer nas organizações, criou ainda mais condições para o desenvolvimento de aplicações cada vez mais exigentes, assentes em modelos integrados de processamento de informação e cada vez mais dependentes de uma infra-estrutura de comunicações fiável, rápida, sem falhas e acessível a partir de qualquer ponto. Por exemplo, as aplicações distribuídas, acessíveis a partir de qualquer ponto através de uma interface Web (*browser*).

Quem agora inicia uma actividade empresarial ligada ao domínio das tecnologias de informação e comunicação (TIC) e dos sistemas de informação (SI), pode não dar conta do que se passou nos últimos vinte anos e do caminho percorrido desde então. Alguns temas, como computação móvel, tecnologias multimédia, redes sem fios, computação ubíqua, capacidades dos discos e da RAM medidas vulgarmente em GigaBytes e velocidades de transmissão em redes locais sempre acima do 100Mbps, fazem parte das tecnologias disponíveis actualmente para as empresas e para cada um de nós. É pois inimaginável pensar que, há apenas vinte anos (para não recuar mais no tempo), a Internet em Portugal ainda vivia a sua infância, as redes sem fios e tudo o que daí surgiu era apenas uma miragem, a capacidade dos discos andava na ordem dos 10 MBytes, a da RAM não iria além do 1 MByte e as comunicações eram pouco fiáveis e com velocidades de transmissão na ordem dos kbps.

O avanço tecnológico registado originou naturalmente o aparecimento de novas profissões ligadas à informática, às redes e à sua gestão e manutenção. Ao longo do tempo, a dependência da infra-estrutura das TIC e dos SI foi aumentando consideravelmente, começando a não se tolerar a "falha no sistema". Tal facto motivou as empresas a contratarem profissionais com perfis técnicos para o diagnóstico e correcção das falhas da rede e dos serviços que lhes estavam associados. Progressivamente, com o aumento crescente das redes e consequentemente da sua complexidade, não só em número de equipamentos (servidores, *switches* e *routers* apenas para referir os mais importantes), mas também na sua heterogeneidade (por exemplo, ao nível das arquitecturas de computadores e sistemas operativos), esses profissionais foram-se especializando em tecnologias específicas, como por exemplo Cisco (maior construtor de routers e *switches*), Microsoft e sistemas Unix.

Concretamente na área de administração e gestão de sistemas e redes, o processo de "especialização" engloba várias camadas de conhecimento e experiência profissional. O leque de sistemas operativos, serviços de rede e topologias em funcionamento numa organização é imenso, obrigando à actualização constante de conhecimentos e à troca de experiências com outros

profissionais da área. Além das bases necessárias sobre o funcionamento das redes e dos sistemas operativos, é fundamental o domínio profundo dos vários conceitos que lhes estão associados, como a gestão e configuração dos serviços de rede, os procedimentos de gestão do sistema operativo, as normas protocolares que são actualizadas quase diariamente (por exemplo, RFC[1] e *drafts* produzidos por grupos de trabalho do IETF) e o impacto que a actualização tecnológica tem no desenvolvimento das novas versões que vão saindo dos sistemas operativos. Todo o trabalho de actualização de conhecimentos e progressão no domínio técnico pode ser conseguido de várias formas, destacando-se a certificação em tecnologias de grande implantação (por exemplo, Microsoft e Cisco), a criação de ambientes laboratoriais para testar novos procedimentos e "truques" úteis, que possam ser aplicados em ambientes de produção, a especialização em várias áreas de actuação e não apenas numa muito restrita e, não menos importante, a iteração com outros profissionais da área e com o meio académico [1-3].

Durante os últimos vinte e dois anos desenvolvi a minha actividade na área de gestão e administração de rede, servidores e serviços. Este período pode ser dividido em duas grandes partes. A primeira, descrita neste relatório, abrange o período entre 1988 e 2001, e foi desenvolvida em ambiente empresarial, como quadro de várias empresas do Grupo Sonae. A segunda, desde 2001, foi desenvolvida como docente do ensino superior politécnico público, como responsável de várias unidades curriculares na mesma área, tendo leccionado aulas teóricas e laboratoriais, bem como orientado vários projectos e estágios. A actividade docente foi integralmente desenvolvida em áreas de conhecimento directamente relacionadas com a experiência profissional previamente adquirida.

Relativamente ao primeiro período, foram quase treze anos, que se iniciaram no final da década de 80, consumiram a totalidade da de 90 e ainda um pouco deste século. Durante esse período estudei várias tecnologias de redes e sistemas operativos, elaborei vários relatórios técnicos e de "boas práticas" e implementei, configurei e geri várias soluções de rede em outros tantos negócios, mais ou menos críticos do ponto de vista do gestor. Durante o mesmo período elaborei e apresentei várias propostas técnicas de soluções no domínio das redes e servidores, apoiei na elaboração de procedimentos de suporte e de salvaguarda de dados, bem como no desenvolvimento de várias aplicações de administração e gestão de servidores e redes. Por fim, tive a meu cargo a gestão e coordenação de equipas de trabalho em vários projectos, alguns importantes e com impacto directo no negócio. Em suma, foram treze anos vividos num ambiente empresarial competitivo, com investimento constante e ambicioso em TIC, com a preocupação constante de disponibilizar ao cliente e, necessariamente, ao utilizador final, um nível de serviço de qualidade.

Este relatório sintetiza a actividade que desenvolvi entre 1988 e 2001 na área de administração de sistemas, redes e serviços de vários negócios do grupo Sonae, maioritariamente da Sonae Indústria e Distribuição. Trata-se de um relatório técnico, do qual não faz parte a apresentação de avanços ou contribuições científicas efectuadas na área em causa. Os principais objectivos do relatório são os seguintes:

[1] Request for Comments (http://www.rfc-editor.org)

1. Descrever o contexto empresarial no período em apreciação.
2. Enquadrar temporal e espacialmente as funções desempenhadas.
3. Descrever as actividades de gestão de redes, servidores e serviços, enquadrando tecnicamente as diversas arquitecturas envolvidas.
4. Descrever os projectos de implementação de soluções de alta disponibilidade recorrendo a tecnologias de clusters de servidores, dando ênfase às soluções para AIX e Linux.
5. Detalhar os projectos mais relevantes e respectivas responsabilidades na definição da solução tecnológica adoptada e na coordenação da sua implementação, nomeadamente:
    a. O Tradutor de EDI (TEDI);
    b. A Nova Plataforma Operacional para os entrepostos da Sonae Distribuição (NPO-Entrepostos);
    c. A implementação de uma solução centralizada de cópias de segurança (*backups*) para os servidores das empresas do universo Sonae.
6. Resumir a actividade desenvolvida no período posterior a 2001, nomeadamente os projectos em colaboração com empresas e a interligação com a experiência profissional adquirida anteriormente.

Embora a função de administração e gestão de sistemas e redes possa considerar-se uma área de especialização *per si*, foram as seguintes as sub-áreas mais relevantes onde adquiri experiência profissional:

1. Instalação, configuração, gestão e administração de servidores IBM OS/36 e OS/38, respectivamente para as arquitecturas S/36 e S/38.
2. Instalação, configuração, gestão e administração de servidores VAX-VMS.
3. Instalação, configuração, gestão e administração de servidores Unix, designadamente AIX, HP-UX e Linux.
4. Desenvolvimento de programas em Unix *shell scripting* e Perl para automatização de tarefas de administração de sistemas e redes.
5. Desenho, implementação e configuração de soluções integradas de alta disponibilidade recorrendo a tecnologias de clusters nos sistemas operativos AIX e Linux
6. Gestão e coordenação de projectos de índole tecnológica.

1.1 CONTEXTO EMPRESARIAL

Falar de Sonae é, de imediato, falar de "Continente". De facto, a face mais visível do grupo Sonae são as lojas de retalho (alimentar e não alimentar) espalhadas por todo o país. Esta constatação é válida, não só pelo volume de negócios que representa e pela quantidade de colaboradores que comporta, mas principalmente pela expansão que tem tido nos últimos anos. Desde a inauguração, em 1985, do primeiro hipermercado Continente em Matosinhos, o grupo conta já com 39 lojas Continente e 125 lojas Modelo, para citar apenas as marcas mais

importantes no negócio do retalho base alimentar [4]. Embora represente uma fatia muito importante do volume de negócios, o grupo Sonae não se resume à área da distribuição. Ao longo das últimas duas décadas cresceu igualmente em outras áreas, das quais se destacam a indústria, as telecomunicações e a gestão de investimentos. De seguida resume-se a actividade que desenvolvi no grupo Sonae.

A minha actividade profissional teve início em Novembro de 1988 no centro de informática da unidade fabril de Souselas da empresa S.I.A.F., Sociedade de Iniciativa e Aproveitamentos Florestais S.A. O primeiro ano de actividade foi dedicado essencialmente ao desenvolvimento de pequenas aplicações informáticas de apoio à manutenção e gestão industrial, bem como às áreas de expedição e controlo de qualidade. Desta primeira fase destaco a aquisição de conhecimentos sólidos sobre os processos produtivos envolvidos no fabrico de um produto, bem como a experiência adquirida sobre o modelo de negócio em causa, os processos de gestão e os fluxos de dados envolvidos na empresa e no grupo.

O período inicial de adaptação originou duas consequências óbvias: por um lado a constatação da necessidade de adquirir conhecimentos e experiência no sistema central IBM S/36, onde assentava o processamento de toda a informação do negócio e a interligação com o sistema central da Sonae Indústria; por outro, a profunda convicção de que a margem de progressão dentro do grupo poderia ser grande. Nesse sentido, encetei uma nova etapa que se traduziu na integração na equipa de gestão do sistema IBM S/36, destacando como principais funções a operação e gestão do sistema e o desenvolvimento de algumas aplicações em RPGII e RPGIII.

Posteriormente, em Outubro de 1990 surgiu a oportunidade de integrar a direcção responsável pelo centro de informática da Sonae Indústria, no complexo industrial da Maia. Nessa altura a actividade da Sonae Indústria era já bastante intensa, traduzida nas sucessivas aquisições de empresas do sector, como por exemplo a Casca (Serém-de-Cima), a Novopan (Rebordosa) e a Paivopan (Castelo de Paiva). A experiência adquirida anteriormente na unidade fabril de Souselas foi fundamental para iniciar uma carreira no centro da Maia, dando-me a possibilidade de assumir desde início a responsabilidade pela gestão e administração do sistema central IBM S/38 e da infra-estrutura de comunicações com os sistemas IBM S/36 das unidades fabris. Além das tarefas de gestão do sistema, assumi ainda a interligação com o departamento de desenvolvimento de aplicações.

Até ao início de 1995 fui responsável operacional das actividades do centro de informática da Sonae Indústria, tendo pertencido aos quadros das empresas Sonae Indústria e Revestimentos, S.A. (S.I.R.) e Sonae Produtos e Derivados Florestais, S.A. (S.P.D.F., S.A.). Durante esse período houve dois marcos profissionais importantes, correspondentes a duas migrações do sistema de informação da Sonae Indústria. A primeira ocorreu sensivelmente em 1992 para uma rede DECnet constituída por vários servidores VAX com sistema operativo VMS. A segunda decorreu em 1995 e consistiu na migração para servidores Unix, mais concretamente para um servidor HP PA-RISC, com sistema operativo HP-UX. Do ponto de vista profissional, estas duas migrações permitiram a aquisição de conhecimentos sólidos sobre duas tecnologias distintas e, acima de tudo, sobre o planeamento e execução dos processos inerentes à migração de um sistema de informação como o da Sonae Indústria. Nesse sentido, destaca-se o planeamento da migração das aplicações, a definição de processos de transição a serem adoptados pelos utilizadores, a definição de

estratégias de contingência e a formação dos profissionais envolvidos na monitorização e operação dos sistemas.

Em meados da década de 90 a Sonae inicia uma fase de expansão na área das telecomunicações e serviços, tendo surgido a empresa Sonae Redes de Dados, S.A. (SRD) pertencente à sub-holding Sonae.com. A SRD foi fundada em 1994 e participada na totalidade pela Sonae Tecnologias de Informação, S.A. (Grupo INPARSA). Os seus principais objectivos consistiam na prestação de serviços de comunicações e integração de tecnologias de informação, contribuindo para que os sistemas de informação dos seus clientes, maioritariamente empresas do grupo Sonae, funcionassem da forma mais adequada às exigências de cada negócio. Os principais serviços disponibilizados incluíam voz, dados, consultoria e projectos, suporte e gestão de sistemas, administração de bases de dados, formação e suporte ao negócio (*help desk*). A área da consultoria e projectos implementava as soluções de integração das várias tecnologias de informação e comunicações, bem como de toda a infra-estrutura tecnológica necessária. Os serviços de suporte e gestão eram prestados nas áreas de microinformática e servidores multi-plataforma. A administração das bases de dados tinha a seu cargo a análise de desempenho e a gestão dos sistemas gestores de bases de dados em várias plataformas de sistemas operativos. O serviço de *help desk* assegurava, a nível nacional, o atendimento telefónico dos clientes e actuação em primeira linha. Em função do tipo e hora da ocorrência, a intervenção de 2ª linha às diversas áreas poderia ser accionada. Os serviços de voz e dados tinham a seu cargo a implementação de soluções de comunicações através de circuitos ponto a ponto de diversos protocolos, nomeadamente, X.25 e *frame relay*. A SRD foi constituída para assegurar os serviços de gestão e administração das infra-estruturas de servidores e redes das várias empresas do grupo, tendo absorvido grande parte dos quadros ligados à gestão de sistemas e redes das empresas da Sonae Indústria e Sonae Distribuição. A SRD foi igualmente a detentora da licença para o terceiro operador móvel de telecomunicações atribuída pela Anacom, que mais tarde seria explorada pela empresa Optimus.

Assim, sensivelmente entre meados de 1995 e final de 1998 desenvolvi a minha actividade na empresa SRD, predominantemente na gestão e administração de servidores e de redes. Ao conhecimento e experiência adquiridos no domínio de sistemas Unix da Sonae Indústria foi adicionada a realidade da Sonae Distribuição que incluia como principal cliente a Modelo-Continente Hipermercados, S.A. (MCH). A partir de meados de 1995 desenvolvi a minha actividade integralmente na área da gestão e administração de sistemas Unix. Dos sistemas de vários clientes da Sonae Indústria e Sonae Distribuição que tive a meu cargo, especializei-me na arquitectura IBM SP (ou simplesmente, SP), composta por um conjunto de vários nós de processamento interligados por um *switch* de elevado desempenho, que partilhavam discos externos interligados por uma rede de alto débito. A uma arquitectura deste tipo, cujos nós partilham um conjunto de discos externos, designa-se por "cluster". Este termo será igualmente usado ao longo do relatório para designar um conjunto de servidores que implementam mecanismos de alta disponibilidade através da retoma automática de recursos. O *switch*, tecnologia proprietária da IBM, constituía a mais-valia do SP quando comparado com outras arquitecturas da mesma gama, permitindo uma comunicação rápida entre os nós e a definição de cenários de processamento paralelo e distribuído. O SP estava inteiramente dedicado aos negócios da Sonae Distribuição, em especial às aplicações de retalho. Inicialmente tinha apenas 5 nós e na

altura era um dos poucos clusters SP na península Ibérica. Começou por alojar a *datawarehouse* da MCH, tendo-lhe sido gradualmente adicionados outros nós, alocados exclusivamente a outras aplicações de negócios da Sonae Distribuição. Em 2001 o SP tinha quatro *frames* completamente preenchidas, tendo-lhe sido igualmente adicionados vários nós externos.

A experiência em sistemas Unix e redes adquirida na Sonae Indústria foi fundamental para encetar esta nova etapa, mais ambiciosa e exigente. As aquisições e fusões constantes levadas a cabo pela Sonae na área da distribuição, bem como a expansão dos negócios já existentes, como por exemplo o número de lojas Modelo e Continente, permitiram a integração em diversos projectos de planeamento, desenho e instalação de várias soluções tecnológicas de rede baseadas em Unix. Tratando-se de negócios críticos para a Sonae, um dos aspectos fundamentais a ter em conta no desenho e implementação destas soluções consistia na utilização de estratégias de tolerância a falhas e alta disponibilidade das aplicações de negócio e dos serviços de rede. Nesse sentido, participei na implementação de vários projectos deste tipo recorrendo a tecnologias de *clustering*, alguns dos quais descritos ao longo deste relatório.

O crescimento dos negócios da Sonae.com levou a várias reorganizações, sendo uma delas a criação da empresa Novis. Directamente ligada à exploração de um serviço de rede fixa de voz, a Novis agregou também a actividade desenvolvida pela SRD. Estava inicialmente dividida em dois grandes grupos: Novis Telecomunicações, responsável pela infra-estrutura de comunicações do grupo Sonae e pela exploração da rede "1010"; Novis Soluções, dedicada integralmente à gestão da infra-estrutura de rede e serviços das empresas do grupo. Paralelamente, a Novis Soluções desenvolveu também o planeamento de novas soluções, a gestão do centro de informática, a consultoria técnica, os serviços de *hosting* e a operação continua (24x7) de toda a infra-estrutura tecnológica do grupo. Foi na Novis Soluções que desenvolvi a minha actividade até Março de 2001, integrando a direcção de planeamento e projectos. Durante esse período exerci a função de gestor sénior de projecto para a área de sistemas Unix e redes. Os projectos que integrei incluiam essencialmente negócios críticos ligados à Sonae Distribuição, onde a componente de alta disponibilidade e tolerância a falhas era fundamental. Dos vários projectos em que estive directamente envolvido destaco o projecto TEDI (Capítulo 4), a instalação da nova plataforma operacional dos entrepostos da Sonae Distribuição (Capítulo 6) e a instalação de uma solução integrada de *backups* baseada na arquitectura ADSTAR Distributed Storage Manager (ADSM) - Tivoli Storage Manager, com uma unidade robotizada de gestão de cópias de segurança (Capítulo 7). Não menos importante foi a participação na migração do centro de informática da SRD para as instalações definitivas da Novis, tendo assegurado a gestão do projecto para a componente de servidores Unix.

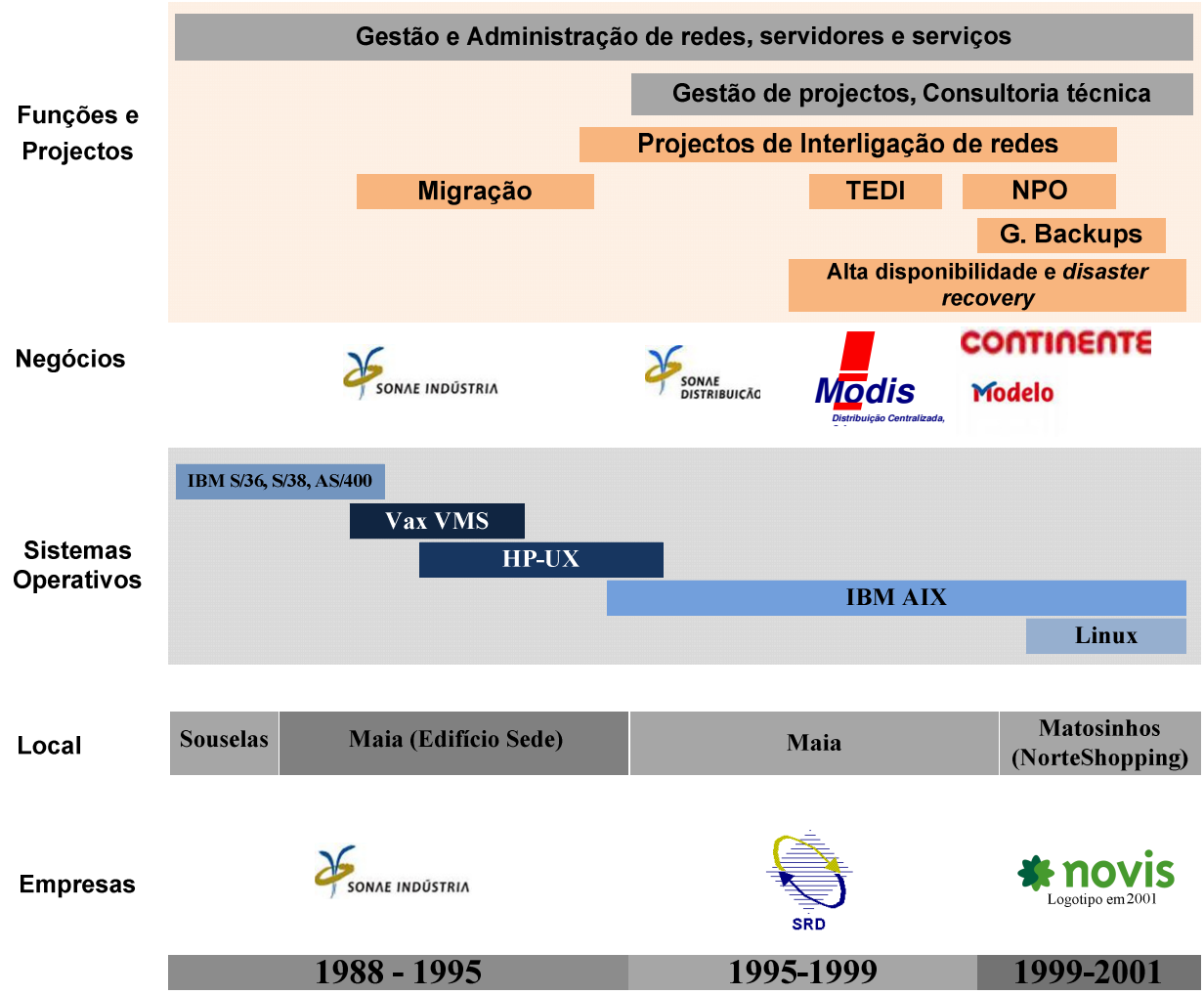


Figura 1-1 – Resumo da actividade desenvolvida.

A Figura 1-1 resume temporalmente a actividade desenvolvida a cinco níveis: empresas e respectiva localização física, sistemas operativos e arquitecturas envolvidas, negócios e projectos relevantes.

1.2 ESTRUTURA DO DOCUMENTO

Este relatório sintetiza as principais actividades desenvolvidas no período atrás referido, realçando as seguintes três sub-áreas de especialização: administração e gestão de sistemas e redes; desenho, configuração e implementação de soluções de alta disponibilidade utilizando tecnologias de *clustering* em sistemas Unix (AIX e Linux), onde se destacam o projecto TEDI; gestão de projectos, nomeadamente o NPO-Entrepostos e a implementação de uma solução centralizada de *backups*. A estrutura do relatório é a seguinte:

- O Capítulo 2 apresenta um enquadramento das redes geridas durante o período em análise, evidenciando as topologias de rede e os tipos de equipamentos envolvidos.

- No Capítulo 3 enquadra-se a problemática da alta disponibilidade e o impacto da sua utilização nas organizações. Descrevem-se ainda as acções envolvidas no desenvolvimento de um cluster de alta disponibilidade.
- No Capítulo 4 descreve-se o projecto TEDI (tradutor de EDI), designadamente a implementação de uma solução de alta disponibilidade usando a aplicação de *clustering* HACMP para AIX.
- O Capítulo 5 apresenta uma solução *open source* de *clustering* de alta disponibilidade para Linux usando a aplicação "heartbeat". Esta solução foi experimentada na rede laboratorial da Novis, tendo permitido identificar as potencialidades associadas ao "heartbeat" e à sua aplicabilidade em cenários empresariais.
- O Capítulo 6 é dedicado ao projecto NPO-Entrepostos. Neste projecto assumi o papel de gestor de projecto, tendo coordenado a implementação da infra-estrutura tecnológica.
- No Capítulo 7 descreve-se o projecto de implementação de uma solução integrada de *backups* para o universo de servidores e aplicações geridas à data pela Novis. Este projecto foi desenvolvido em paralelo com o projecto NPO, onde assumi a consultoria técnica para a integração dos servidores Unix na infra-estrutura centralizada de *backups*.
- No Capítulo 8 sintetizam-se as principais conclusões à actividade desenvolvida durante o período em análise. Genericamente, faz-se um balanço da experiência adquirida, descreve-se o ponto de situação actual das soluções implementadas e faz-se a ligação com o período posterior a 2001.
- Após a bibliografia, são apresentados vários documentos complementares ao relatório.

2 ADMINISTRAÇÃO DE SISTEMAS E REDES

A função de administração de servidores, redes e serviços foi por mim desempenhada em várias arquitecturas de computadores com sistemas operativos e tecnologias de redes distintas. Nesta secção descrevem-se sumariamente as redes e servidores da Sonae Indústria, nomeadamente IBM (IBM S/38 e IBM S/36), VAX-VMS e da migração para Unix, assente sobre servidores HP com sistema operativo HP-UX. Descreve-se igualmente a infra-estrutura IBM SP da Sonae Distribuição, realçando os seus componentes tecnológicos e as aplicações envolvidas.

2.1 IBM S/36 E IBM S/38

A Figura 2-1 ilustra a rede da Sonae Indústria em 1990, constituída por sistemas IBM S/36 e S/38 [5-8]. A rede continha quatro nós principais, designadamente Maia, Souselas, Mangualde e Oliveira do Hospital. Aos nós da Maia e de Souselas estavam definidas ligações a unidades fabris mais pequenas, como Castelo de Paiva ou Rebordosa. O servidor da Maia incluía o SI das empresas existente no pólo da Maia, designadamente SIR, SPDF e Poliface. A configuração da rede foi sofrendo algumas alterações, fruto da aquisição de algumas unidades fabris, onde se colocava um novo servidor dimensionado às necessidades de processamento e ao número de utilizadores.

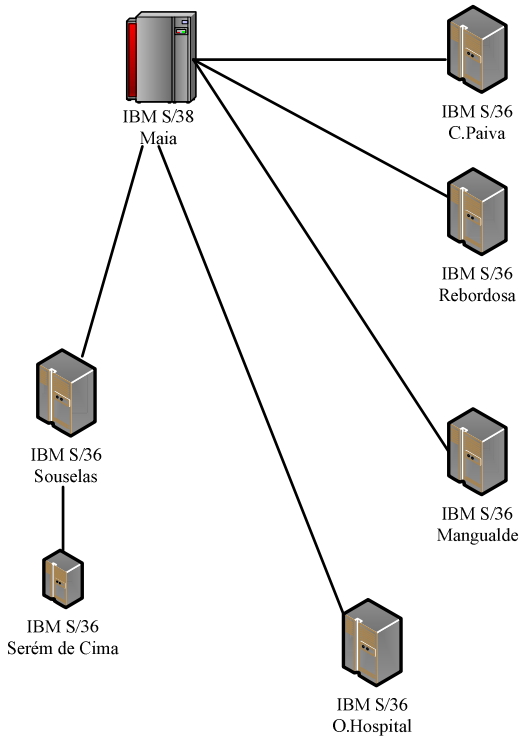


Figura 2-1 – Rede de servidores IBM da Sonae Indústria.

A rede tinha apenas um servidor IBM S/38 nas instalações da Maia, sendo os restantes do modelo S/36. As diferenças eram essencialmente na configuração física e no sistema operativo suportado. Embora muito semelhantes, os sistemas operativos OS/36 e OS/38 continham algumas

diferenças relativamente ao modelo de gestão de alguns subsistemas (por exemplo, disco e serviço de impressão).

As ligações entre os vários sistemas eram X.25, havendo algumas do tipo permanente (PVC) e outras comutadas quando necessário (SVC). Por exemplo, as ligações às unidades fabris mais pequenas eram normalmente do tipo SVC. Cada servidor suportava um conjunto vasto de aplicações de gestão da unidade fabril em causa, nomeadamente *stocks*, facturação, contabilidade, gestão de produto e expedição. Os programas estavam codificados em RPGII no S/36 e RPGIII no S/38.

2.2 VAX-VMS

A Figura 2-2 ilustra a rede VAX/VMS da Sonae Indústria em 1993. A rede continha três nós principais, correspondentes às unidades fabris mais importantes: Maia, Oliveira do Hospital e Mangualde. Em cada um dos servidores existiam ligações às unidades fabris de menor dimensão, assegurando a ligação remota dos utilizadores.

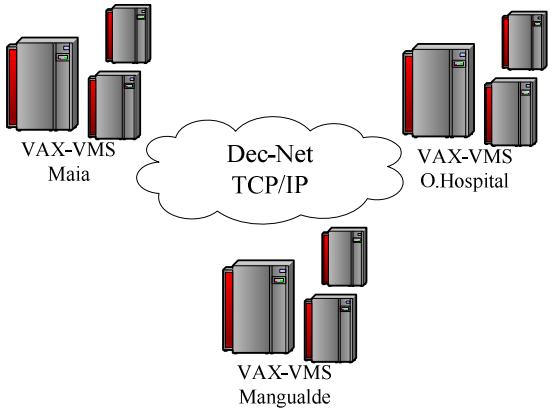


Figura 2-2 – Rede de servidores VAX-VMS da Sonae Indústria.

A arquitectura VAX foi desenvolvida pela DEC e constituiu uma das primeiras a integrar o conceito de cluster, através da partilha de recursos de disco por vários servidores [9]. Os servidores VAX usavam um sistema operativo proprietário, o VMS (Virtual Management System) [10] e estavam interligados por uma pilha protocolar de rede igualmente proprietária, a DECnet [11].

2.3 HP-UX

O PA-RISC é uma arquitectura proprietária da HP que suporta o sistema operativo HP-UX, uma variação do sistema Unix baseado no System V [12]. A Figura 2-3 ilustra a rede da Sonae Indústria com servidores HP-UX. É muito semelhante à rede VAX-VMS, tendo sido instalados apenas três servidores HP-UX nas unidades fabris principais: Maia, Mangualde e Oliveira do Hospital.

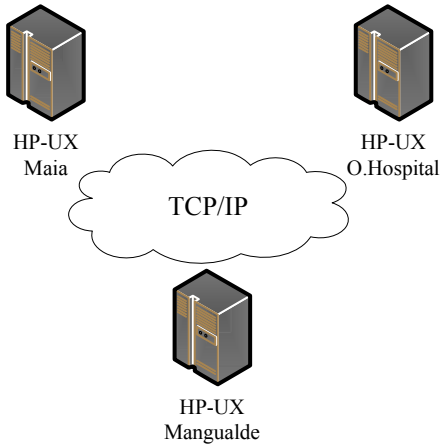


Figura 2-3 – Rede de servidores HP-UX da Sonae Indústria.

Entre 1988 e 1994 a rede da Sonae Indústria conheceu três configurações distintas, assentes em três configurações proprietárias: o IBM S/36 e S/38, o VAX-VMS e o HP-UX. Ao nível da rede, foram usadas igualmente três pilhas protocolares distintas: o SNA para IBM [13], a DECnet para VAX-VMS e o TCP/IP. A Figura 2-4 ilustra essa migração.

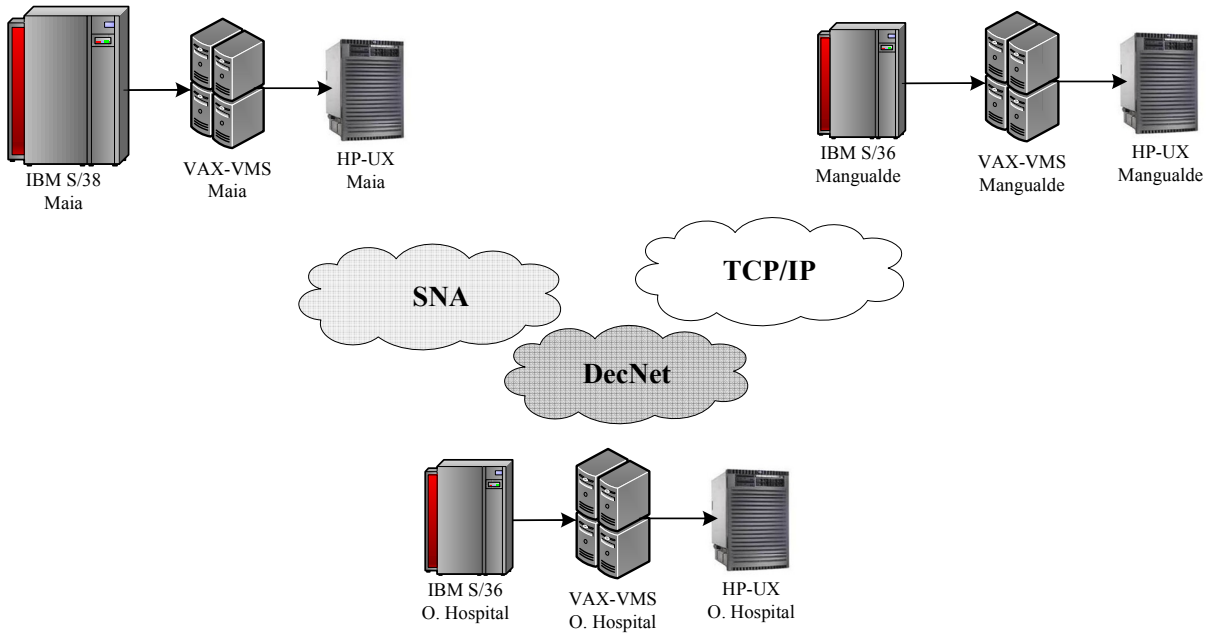


Figura 2-4 – Visão global das migrações levadas a cabo na Sonae Indústria.

Assim, durante esse período foi necessário realizar dois processos de migração do sistema de informação. A alguns níveis, a transformação foi bastante profunda, implicando a substituição de componentes chave. Por exemplo, nas arquitecturas proprietárias IBM (S/36 e S/38) o sistema gestor de base de dados (SGBD) era o DB2, proprietário da IBM [14], enquanto no servidor HP-UX, o SGBD usado passou a ser o Oracle. As aplicações foram igualmente reformuladas e houve uma revisão profunda dos processos de negócio implementados até à data.

2.4 IBM SP

As principais aplicações de gestão do negócio da Sonae Distribuição residiam no super computador IBM Scalable Power (SP). As unidades de processamento do SP, designadas por "nós", baseavam-se na arquitectura RISC/6000 com o sistema operativo AIX e estavam alojados em armários específicos (*frames*). A Figura 2-5 ilustra a rede global do SP em 2001. Era composta por quatro *frames* completas e três nós externos (modelos H50, S7A e S70). Os nós dispunham de ligações aos discos externos através do protocolo Serial Storage Architecture (SSA), proprietário da IBM. Os nós eram maioritariamente do tipo Symmetric Multi-Processing (SMP), podendo contudo serem usados para processamento paralelo, adoptando assim uma postura de Massive Paralell Processing (MPP) [15, 16].

O SP possuía ainda algumas características distintivas, quando comparado com outras arquitecturas, nomeadamente o facto de possuir um *switch* de elevado desempenho a interligar todos os nós [17], bem como as potencialidades de controlo centralizado do sistema através de um conjunto de programas de administração, denominado Parallel System Support Programs (PSSP), igualmente proprietário da IBM [18].

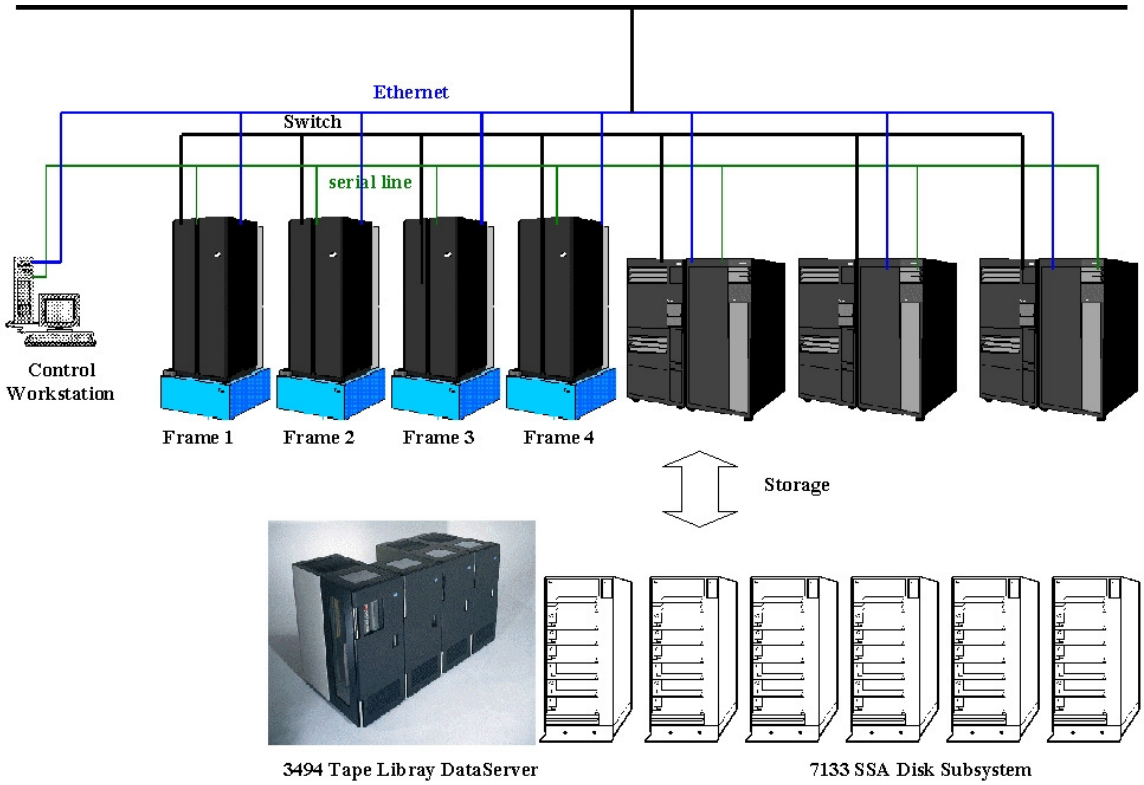


Figura 2-5 – Visão global do cluster IBM SP da Sonae Distribuição.

Assim, os principais componentes básicos de uma arquitectura SP eram:

- **Nós de processamento**: Constituídos por uma unidade de processamento e memória, um ou mais discos internos e adaptadores para ligação a discos SSA externos. Os nós eram da arquitectura RISC/6000, podendo cada um ocupar um número de *slots* diferentes na *frame*. Os vários nós do SP definiam um cluster, podendo partilhar um mesmo conjunto de discos

externos. Cada nó dispunha igualmente de uma interface Ethernet para ligar à rede interna do SP e ainda um adaptador para ligar ao *switch* interno do SP. Opcionalmente, poderia ainda conter diversos adaptadores para outros periféricos, como interfaces de rede Ethernet externas ao SP e adaptadores assíncronos.

- ***Switch* de elevado desempenho (*High-Performance Switch*)** [17]: Este componente correspondia à mais-valia disponibilizada pela IBM para o SP. Através deste *switch* tornava-se possível efectuar ligações entre os nós utilizando uma rede de elevado desempenho. Por exemplo, a partilha de recursos de disco pelo protocolo NFS (ou outro sistema gestor de ficheiros de rede) entre os nós do SP tinha um desempenho muito superior quando comparado com a rede Ethernet. O *switch* tinha ainda a possibilidade de manter actualizadas as rotas entre todos os nós existentes no SP, podendo seguir sempre o melhor caminho entre qualquer par de nós. Se um qualquer caminho não estivesse disponível, a ligação entre os nós seria realizada por um outro alternativo.
- **Ethernet interna do SP**: Existia na configuração base do SP e servia para a comunicação entre os nós. Os endereços IP utilizados eram naturalmente privados e apenas visíveis no interior do cluster.
- **Estação de controlo**: Correspondia a mais uma unidade de processamento, normalmente um PC, mas fora da *frame* onde se encontravam os nós. A sua principal função consistia na gestão e controlo centralizado dos nós do SP. Cada *frame* continha uma ligação série à estação de controlo, através da qual eram enviadas constantemente informações sobre o estado de todos os nós. Através de utilitários de gestão era assim possível controlar e intervir na resolução de eventuais problemas. Algumas tarefas de gestão dos nós eram realizadas a partir da estação de controlo, sendo o acesso iterativo a cada nó realizado em casos excepcionais.

A Figura 2-6 ilustra uma arquitectura possível do SP com duas *frames*, onde é possível identificar os principais componentes descritos anteriormente.

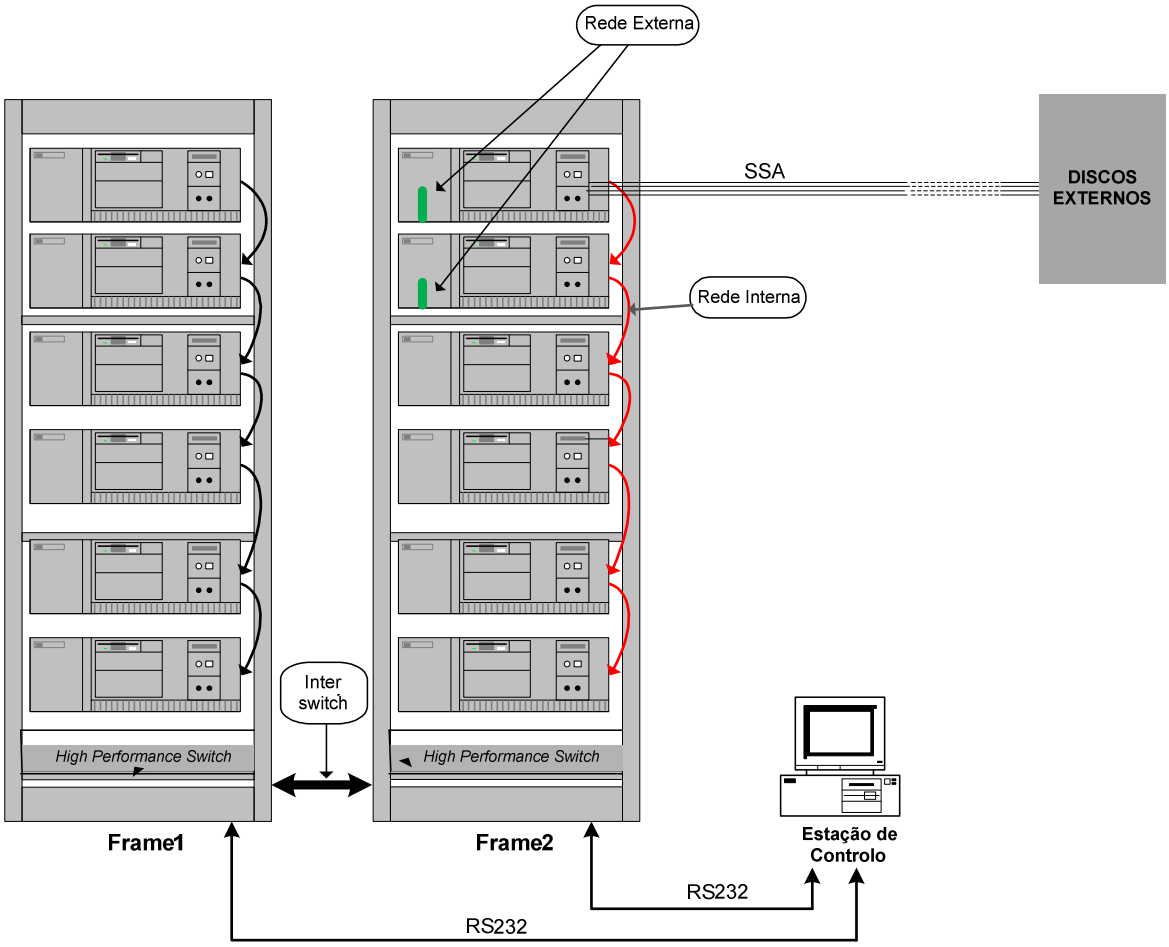


Figura 2-6 – Componentes principais da arquitectura IBM SP.

Em termos de redundância, o SP implementava vários conceitos intrínsecos. Por exemplo, as *frames* tinham incorporadas fontes de alimentação e sistemas de ventilação redundantes. O protocolo SSA assegurava igualmente mecanismos de redundância de caminhos no caso de um disco estar inacessível.

Na arquitectura do SP havia dois nós dedicados exclusivamente a testes de configurações. Das várias experiências realizadas destacam-se as relacionadas com as estratégias de partilha de recursos, em especial a utilização de tecnologias de distribuição de *filesystems* pelos nós do SP, designadamente os testes realizados com o General Parallel File System (GPFS) [19].

O GPFS para AIX permite o acesso global aos ficheiros por todos os nós de um cluster que tenham configurado aquele tipo de *filesystems*. O GPFS é um *filesystem* paralelo que oferece vantagens a nível de desempenho, ao eliminar a limitação de existir um só servidor para disponibilizar o acesso a ficheiros, como também ao nível da flexibilidade. Com um *filesystem* paralelo, uma vez que todos os nós podem aceder ao *filesystem*, é mais fácil mover aplicações de um nó para outro. Este facto é especialmente válido para soluções de alta disponibilidade, como o HACMP (Capítulo 3), onde a aplicação é retomada entre os servidores na ocorrência de erros não recuperáveis no servidor principal. O GPFS permite ainda que os utilizadores tenham acesso

partilhado a ficheiros que podem estar distribuídos por vários discos e configurados em múltiplos nós.

Este tipo de *filesystem* oferece alta escalabilidade e disponibilidade ao permitir que múltiplos servidores e discos tenham acesso ao mesmo *filesystem*. Se um servidor falhar, o *filesystem* continuará disponível desde que o outro servidor tenha acesso aos discos que contêm os dados e um caminho pela rede para o cliente. A Figura 2-7 esquematiza a arquitectura de GPFS instalada.

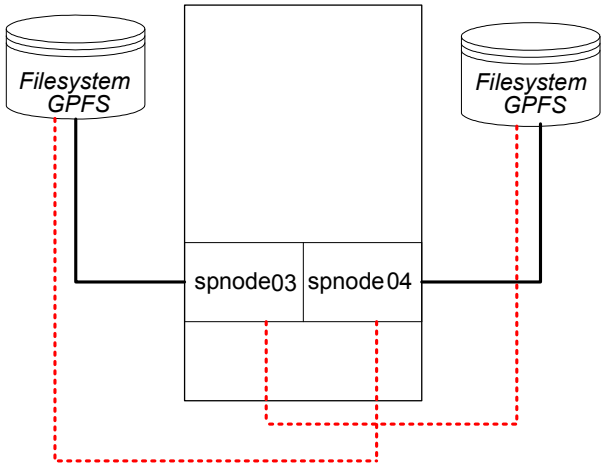


Figura 2-7 – Arquitectura de testes com o GPFS.

Em cada um dos nós do SP foi configurado um *filesystem* partilhado de GPFS. Os testes realizados tinham como objectivo principal comparar o desempenho dos *filesystems* do tipo GPFS e NFS em operações de leitura e escrita. Foi possível verificar que os *filesystems* do GPFS apresentaram um desempenho médio nos processos de cópia na ordem dos 30%, quando comparado com *filesystems* do tipo NFS [20].

3 CONCEITOS DE ALTA DISPONIBILIDADE

O conceito de alta disponibilidade pode ser encontrado em vários serviços que contratualizamos para assegurar o nosso dia-a-dia, como por exemplo o abastecimento de água ou energia eléctrica. Habituámo-nos, ao longo dos tempos, a ter estes serviços sempre disponíveis, de tal forma que não conseguimos admitir a sua paragem, quer planeada ou não. Por exemplo, esperamos que um serviço de abastecimento de energia eléctrica seja altamente disponível, sendo as suas interrupções, até as mais pequenas, incomodas e nalguns casos inaceitáveis. Por outro lado, se existirem interrupções no abastecimento de energia, esperamos que a entidade que disponibiliza o serviço esteja dotada de mecanismos automáticos de recuperação após uma falha, para que o impacto no nosso dia-a-dia seja o mínimo possível.

A contratualização de um serviço pressupõe a existência de uma garantia de nível do serviço, geralmente designado por *Service Level Aggreement (SLA)*. Dependendo do nível de serviço contratualizado, a paragem poderá ser mais ou menos admissível por parte do utilizador. Por outro lado, os utilizadores que dependem do serviço em causa implementam normalmente soluções internas que minimizem as eventuais paragens não planeadas. Por exemplo, algumas organizações dispõem normalmente de geradores que asseguram o funcionamento das máquinas principais durante o período de tempo em que haja interrupção do serviço.

Nas organizações modernas, as infra-estruturas de rede são essencialmente compostas por servidores e equipamentos activos de rede (por exemplo, routers e *switches*), que também necessitam de assentar em topologias que implementem mecanismos de tolerância a falhas e de alta disponibilidade dos serviços que suportam. Para assegurar o funcionamento da infra-estrutura em caso de falha, a tecnologia necessária pode ser mais ou menos complexa em função do negócio. Por exemplo, um negócio assente em tecnologias de *e-commerce* necessita que as aplicações Web estejam sempre disponíveis e com um tempo de resposta aceitável. Caso contrário, poderão perder-se clientes e consequentemente receitas.

Há várias formas de obter arquitecturas com alta disponibilidade, dependendo da criticidade do negócio e do orçamento disponível. Relativamente às soluções de alta disponibilidade recorrendo a clusters de servidores, os conceitos relacionados são muito semelhantes entre as várias arquitecturas de computadores e de sistemas operativos, havendo normalmente variações específicas que estão normalmente relacionadas com as tecnologias de hardware e software envolvidas.

Os conceitos gerais apresentados neste capítulo encontram-se diluídos em várias referências da literatura sobre o tema, destacando-se [21-25]. Assim, este capítulo pretende sumarizar os mais importantes no âmbito da temática da alta disponibilidade em redes empresariais, sendo igualmente apresentadas algumas consequências derivadas da implementação deste tipo de soluções nas organizações. Estes conceitos pretendem suportar a descrição dos dois casos de estudo apresentados nos Capítulos 4 e 5, que correspondem à experiência adquirida por mim na implementação de várias soluções deste tipo em redes de negócios críticos da Sonae Distribuição.

3.1 NOÇÃO DE DISPONIBILIDADE

O termo "disponibilidade" (*availability*) representa o período de tempo em que um sistema ou uma rede estão disponíveis e com um tempo de resposta dentro dos níveis contratualizados com o utilizador. Ou seja, no cálculo da disponibilidade de um sistema não deve apenas questionar-se se uma aplicação está disponível, mas sim se todo o sistema está a fornecer os seus serviços aos utilizadores a um nível adequado. Uma quebra de serviço, planeada ou não planeada, ou ainda uma degradação do tempo de resposta para valores não aceitáveis pelo utilizador, provoca um período de tempo de inacessibilidade (ou *downtime*), medido em unidades de tempo [21].

O valor da disponibilidade de um sistema não é um atributo mensurável em tempo real, como a velocidade de processamento do CPU ou da rede. A disponibilidade de um sistema é normalmente calculada a partir de dados históricos baseados no registo dos períodos e duração das paragens ocorridas. Estes registos podem realizar-se individualmente para cada componente (de *hardware* ou aplicação) da infra-estrutura, ou para o sistema ou rede. Assim, por um lado pode haver interesse em saber apenas qual a disponibilidade efectiva das unidades de armazenamento de dados (por exemplo, discos) ou do servidor responsável pelo correio electrónico. Por outro, pode simplesmente efectuar-se o registo das paragens globais do sistema, independentemente do componente que a causou. Em termos gerais, quer para um componente individual ou para o servidor ou rede, o valor da disponibilidade refere-se à percentagem da sua actividade, calculado pela seguinte fórmula:

$$d = (T - D)/T$$

Na fórmula, T corresponde ao período de actividade e D corresponde ao período em que um determinado componente esteve inactivo, causando indisponibilidade do serviço para o utilizador. Esta indisponibilidade pode ser causada tanto por uma paragem (planeada ou não), como por uma degradação do tempo de resposta. Ambos os valores estão expressos em unidades de tempo. Por exemplo, considerando que a disponibilidade máxima atingível por um serviço numa semana é de 168 horas (24x7), se houver uma paragem de quatro horas, então a disponibilidade atingida cifra-se nos 97.61%. Esta medida estabelece assim a probabilidade de um sistema estar disponível para o utilizador. É, normalmente, expresso como uma percentagem de horas por semana, mês, ou ano durante o qual o sistema e os seus serviços podem ser usados normalmente.

A média de indisponibilidade é calculada pelo período de tempo em que um componente está indisponível, devido a um evento de falha que provoque quebra do serviço para o utilizador, e é calculada pela seguinte fórmula:

$$i = TTI/NF$$

Neste caso, *TTI* representa o tempo total de indisponibilidade e *NF* o número de falhas observadas durante o período de tempo em análise.

A disponibilidade está também relacionada com a percentagem de falhas dos componentes individuais do sistema. O tempo médio entre falhas (TMEF) calcula o tempo médio entre duas falhas num mesmo componente, por exemplo discos, e é calculada da seguinte forma:

$$TMEF = TTO/NF$$

Neste caso, *TTO* refere-se à soma dos tempos de operação de todos os componentes, incluindo aqueles que não falharam, sendo *NF* a soma de todas as falhas em todos os componentes. Em termos práticos, o *TMEF* é usado como um valor esperado do desempenho dos vários componentes do sistema e permite, entre outros aspectos, aferir sobre a próxima falha de um desses componentes.

O tempo médio de reparação (*TMR*) de um componente do servidor representa o tempo gasto na sua reparação. Assim, a medida de disponibilidade de um componente pode igualmente ser calculada através dos tempos médios entre falhas e de reparação, usando a seguinte fórmula:

$$d = TMEF/(TMEF + TMR)$$

É comum distinguir as medidas de disponibilidade para os componentes individuais de um sistema e para o serviço disponibilizado. Assim, o TMEF refere-se aos componentes e o tempo medido entre quebra de serviço[2] (TMEQS) refere-se ao serviço global prestado pelo servidor. A mesma abordagem é seguida por alguns autores para outras medidas de disponibilidade [24].

3.2 TIPOS DE PARAGEM DO SISTEMA

Um sistema pode parar por diversas razões, deixando os seus serviços indisponíveis. Essas paragens podem ser planeadas ou não planeadas, originando sempre um período de paragem (*downtime*), que provoca o desagrado e, por vezes, a desconfiança, do utilizador final face aos sistemas que suportam as aplicações. Por esse motivo, as paragens deverão ter sempre uma duração tão curta quanto possível e ser, sempre que possível, justificadas.

As paragens planeadas estão previstas pela equipa de gestão do sistema e pressupõem um aviso prévio a todos os utilizadores. Alguns exemplos são:

- Operações de salvaguarda dos dados que impliquem paragem das aplicações.
- Instalação de uma nova versão do sistema operativo ou de aplicações (por exemplo, do sistema gestor de base de dados).
- Expansão ou reparação de componentes de *hardware*.
- Alterações diversas à configuração lógica ou física do sistema.

As paragens não planeadas surgem por causas inesperadas e, por vezes, não há qualquer possibilidade de avisar os utilizadores. Este tipo de paragens tem normalmente um maior impacto negativo nos utilizadores, podendo ser causadas por:

- Falha de um componente de *hardware*.
- *Filesystem* a 100% de ocupação.
- Falha de corrente eléctrica.
- Problemas na rede local que impossibilita o acesso dos utilizadores ao sistema.
- Desastres provocados por causas naturais (incêndio, inundação, etc.)

[2] Corresponde à medida Mean Time Between Service Outage (MTBSO), adoptado por alguns autores.

- Erro humano.

Segundo vários relatórios, os erros humanos correspondem à causa mais frequente de paragens não planeadas, sendo normalmente provocados por dois tipos de razões: falta de cuidado na execução dos comandos enviados ao sistema operativo e pouco conhecimento técnico sobre a forma como os vários componentes do sistema funcionam [21].

3.3 NÍVEIS DE DISPONIBILIDADE

Alta disponibilidade, ou *high availability*, designa um modelo possível de ser implementado numa organização, de forma a tornar os seus serviços altamente disponíveis para o utilizador final. O principal objectivo é a redução dos tempos de paragem e pode ser conseguido a vários níveis, nomeadamente utilizando mecanismos intrínsecos a alguns componentes do servidor.

Um sistema tolerante a falhas tem a capacidade de continuar um serviço apesar da existência de uma falha num componente de *hardware* ou nas aplicações. São sistemas que têm normalmente redundância de vários componentes de *hardware* (por exemplo, discos e subsistema de I/O), cujas réplicas entram em funcionamento imediatamente após a falha do componente principal.

Um conceito mais ambicioso de alta disponibilidade é a disponibilidade contínua e consiste na implementação de arquitecturas que minimizem os tempos de paragem ao máximo. Neste tipo de cenários todo o processamento é replicado em dois servidores distintos, garantindo que em caso de falha de um deles, o outro mantém o estado mais recente e estável dos dados. Esta modalidade é naturalmente mais onerosa, obrigando à duplicação de toda a infra-estrutura de *hardware* e necessariamente dos programas que estão em execução. Tal facto implica a implementação de mecanismos mais exigentes de gestão e administração de toda a infra-estrutura [26].

Os planos de *disaster recovery (DRP)* incluem não só a recuperação dos serviços em caso de falha, mas também de todos os componentes vitais da organização, como as comunicações e os *backups* realizados *offsite* [21, 27]. São obrigatórios em organizações de grande dimensão com responsabilidades perante os accionistas e o objectivo fundamental é definir um plano de recuperação do negócio após um desastre de grandes dimensões, como sejam os desastres naturais ou atentados terroristas. Pretende-se que toda a organização esteja preparada para o plano de contingência global e, num curto espaço de tempo, os dados do negócio fiquem disponíveis para retomar a possível normalidade após uma situação de catástrofe. Há vários centros dedicados a este tipo de serviços a grandes empresas, dos quais destaco o Business Continuity and Recovery Service (BCRS) da IBM [26, 28]. Em Portugal há dois dos cinco centros a operar na península Ibérica e a sua infra-estrutura (física e técnica) assegura o normal funcionamento de uma instituição em caso de destruição total ou parcial das suas instalações, que impliquem inacessibilidade dos utilizadores aos serviços aplicacionais e à rede[3].

A Figura 3-1 representa a evolução do custo e esforço dispendido para alcançar cada um dos níveis de disponibilidade [21].

[3] Em http://www-05.ibm.com/services/pt/its/bcrs/ é possível encontrar informação detalhada sobre os dois centros de BCRS da IBM em Portugal.

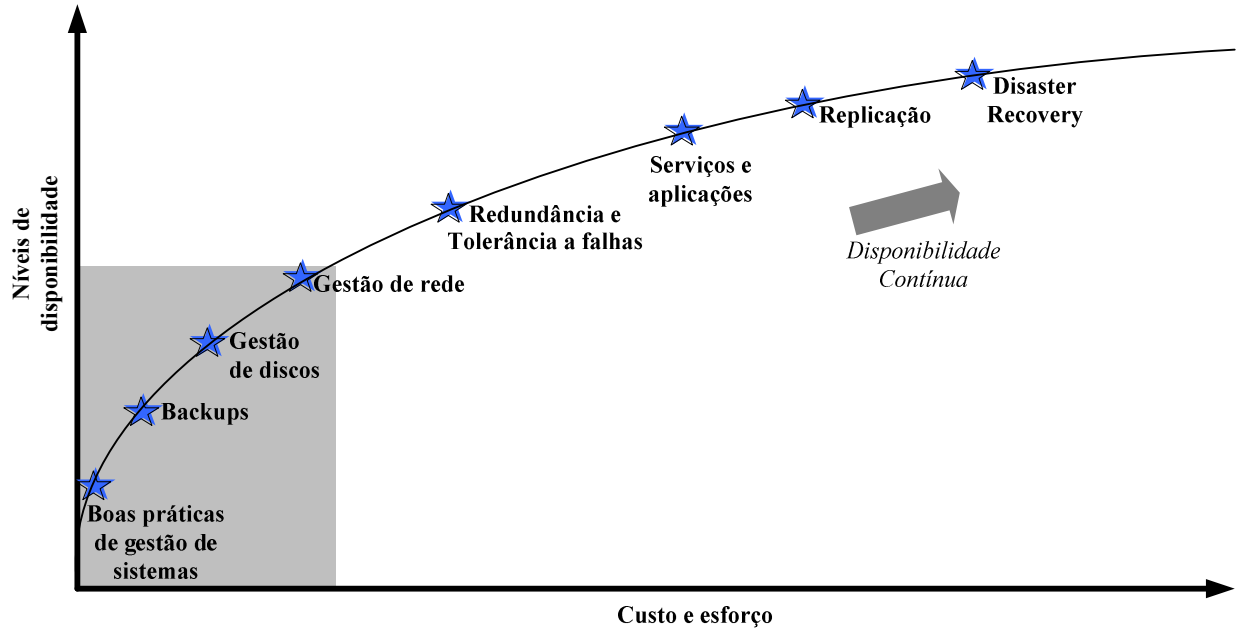


Figura 3-1 – Relação entre os níveis de disponibilidade e o correspondente investimento.

A primeira constatação é de que a implementação de alta disponibilidade começa com a aplicação de boas práticas de administração e gestão dos sistemas e da rede, bem como da definição de boas políticas na gestão das salvaguardas dos dados. Ou seja, explorando as potencialidades de gestão e administração dos servidores e da rede, disponibilizadas pelos sistemas operativos, é possível revestir a infra-estrutura de alguns mecanismos de recuperação de falhas. É igualmente visível na figura que, à medida que se pretende implementar mecanismos mais arrojados de redundância e disponibilidade na arquitectura, o esforço financeiro em *hardware*, *software e* recursos humanos vai aumentando.

A Figura 3-2 (esquerda) ilustra a possível hierarquização em pirâmide dos diversos componentes de um servidor. A duplicação desses componentes pode ocorrer geograficamente a vários níveis, com abrangência local ou não, conforme se apresenta na Figura 3-2 (direita).

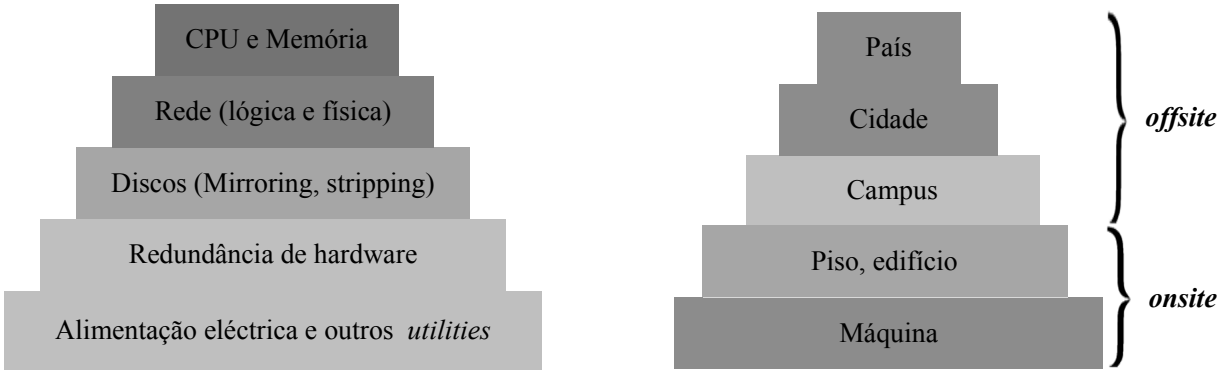


Figura 3-2 – Pirâmide com os principais níveis de disponibilidade.

Os níveis de disponibilidade podem ser alcançados de várias maneiras, dependendo do tipo de negócio envolvido e do orçamento disponível. De seguida apresentam-se alguns exemplos de acções que podem ser realizadas para alcançar alta disponibilidade e redundância a falhas:

- Efectuar redundância apenas das unidades de processamento.
- Utilizar aplicações (incluindo sistemas operativos) que implementem técnicas de mudança automática (e transparente para o utilizador) de utilização de um componente de *hardware* em caso de falha.
- Planear criteriosamente os tempos de paragem.
- Eliminar a interacção humana no sistema, tentando mitigar os erros por essa via.
- Definir mecanismos de resposta automática a condições de erro nas aplicações.
- Definir e utilizar testes de aceitação (aplicação, infra-estrutura) compreensivos e tangíveis, que prevejam um leque alargado de possíveis falhas.
- Definir e praticar resposta a falhas não planeadas e não suportadas por respostas automáticas.

3.4 IMPACTO NA ORGANIZAÇÃO

A implementação de soluções que assegurem a continuidade de serviços implica necessariamente alterações no sei da organização. Essas alterações são visíveis na forma como a organização encara as paragens dos serviços, na forma como lida com as salvaguardas dos dados e ainda na metodologia que utiliza para reagir a desastres, apenas para citar os aspectos mais importantes. Essas alterações estão directamente relacionadas com o nível de disponibilidade que se pretende atingir e podem requerer apenas um ajuste aos processos de monitorização, ou até alterar substancialmente outros aspectos, como o contracto de manutenção que se tem com os fornecedores de componentes de *hardware*.

São várias as oportunidades que um modelo de alta disponibilidade pode trazer para a organização. Embora estejam dependentes de cada área de negócio, pode afirmar-se que os seguintes princípios são normalmente verdadeiros:

- O nível de disponibilidade é determinado pelas necessidades do negócio. Não há um valor absoluto de disponibilidade que possa ser aplicado para todos os negócios.
- Há várias formas de obter disponibilidade da infra-estrutura de TIC e SI e os meios para a alcançar afecta vários aspectos, nomeadamente ao nível do *hardware* adoptado e da forma como as aplicações são desenvolvidas, entre outros.
- O impacto negativo da falha do sistema pode ser reduzido, criando procedimentos claros e acções de manutenção preventiva. Por exemplo, implementado metodologias de treino de todos os intervenientes (equipas de gestão e administração de sistemas, utilizadores e equipas de desenvolvimento).
- A mudança de atitude perante as falhas deve necessariamente abranger toda a organização. Além das rotinas de treino, é desejável a implementação de planos de contingência a

situações de desastre que envolvam todos os departamentos organizacionais, nomeadamente as áreas de gestão do negócio.

- A organização deve ver a utilização de soluções de alta disponibilidade como prioritária, identificando as necessárias oportunidades de negócio que lhes possam estar associadas. Por exemplo, os modelos de negócio baseados em tecnologias de *e-commerce* poderão não sobreviver se não adoptarem soluções deste tipo.

As aplicações de suporte a negócios críticos deverão igualmente estar adaptadas para um ambiente de alta disponibilidade. Os seguintes princípios básicos deverão estar garantidos, não apenas para as aplicações desenvolvidas internamente como também para as adquiridas ao exterior:

- O desenvolvimento das aplicações deve assegurar o mínimo de paragens não planeadas.
- As aplicações devem ter processos de recuperação após as falhas que assegurem sempre a integridade e consistência dos dados. Por exemplo, os mecanismos de recuperação associados ao arranque de um sistema gestor de base de dados após uma paragem imprevista e não planeada.
- As aplicações devem ser configuráveis sem necessidade de realizar uma paragem do sistema para que as alterações tomem efeito.
- As aplicações devem estar munidas de mecanismos auxiliares de monitorização e gestão, como por exemplo procedimentos bem definidos de paragem e arranque.

Um modelo de alta disponibilidade pode ser alcançado a vários níveis. A cada nível corresponde uma área do sistema que se pretende proteger de eventuais falhas. A Tabela 3-1 ilustra esses níveis e descreve como, em cada um deles, se pode atingir uma solução de disponibilidade elevada [23, 24].

Nível	Soluções/estratégias para implementar alta disponibilidade
Cluster	Comunicação entre os nós do cluster deve ser fiável e dedicada. Os dados devem estar armazenados em suportes que assegurem a protecção física e lógica. Deverão ainda existir vários nós associados às aplicações existentes no cluster.
Servidor	Redundância de CPU e de adaptadores de I/O para acesso aos dados.
S.O.	Implementação de *mirroring* dos discos do sistema operativo.
Gestão do sistema e da rede	Adoptar mecanismos de administração centralizada do sistema. Adoptar igualmente ferramentas de monitorização da rede. Implementar mecanismos de automatização de tarefas de gestão.
Base de Dados	As bases de dados deverão ser capazes de arrancar num nó diferente ou estar em execução em dois nós em simultâneo.
Aplicacional	As aplicações têm de ser robustas e efectuar a recuperação após uma falha do sistema. Deverão ser capazes, juntamente com os monitores de transacções, de mudarem para outro sistema, em caso de falha no actual.
Hardware	Deverão ser implementadas técnicas de comutação entre vários componentes redundantes.

Tabela 3-1 – Níveis de disponibilidade e principais estratégias para os alcançar.

De seguida descrevem-se as principais preocupações que a organização deve ter em conta durante o processo de implementação multi-nível de uma solução de alta disponibilidade. Estas preocupações reflectem a minha experiência adquirida no desenvolvimento de algumas soluções deste tipo, nomeadamente as relatadas nos Capítulos 4 e 6.

1. **Definir objectivos concretos e claros.** Deve conhecer-se muito bem as necessidades dos utilizadores e do negócio. Devem igualmente identificar-se muito bem as aplicações de negócio utilizadas e, com base no nível de serviço que se pretende, iniciar a migração para um sistema assente em alta disponibilidade. Um bom ponto de partida é a elaboração de um SLA onde sejam inscritos os níveis de serviço requeridos pelos utilizadores nos períodos normais de operação, o planeamento de algumas paragens e os níveis de desempenho exigidos. Por exemplo:
    - O sistema deverá estar disponível 99.5% numa base 24x5x52 (24 horas por dia, 5 dias por semana e 52 semanas por ano).
    - O tempo de resposta deverá ser de 1.2 segundos para um PC, excepto durante a execução do *backup* diário.
    - Os *backups* incrementais serão realizados uma vez por semana e têm uma duração estimada de 90 minutos.
    - Os *backups* incrementais *online* serão realizados diariamente, provocando um aumento de resposta para 2-3 segundos e não terão uma duração superior a 30 minutos.
    - O tempo de recuperação após uma falha será, no máximo, de 5 minutos.
2. **Definir o ambiente físico apropriado.** Uma solução de alta disponibilidade inclui também o isolamento de eventuais problemas relacionados com energia eléctrica, condições de temperatura, de humidade e da cablagem diversa.
3. **Automatizar processos.** Do ponto de vista da administração e gestão do sistema é fundamental desenvolver procedimentos automáticos (por exemplo, através de Unix *shell scripts*) para o maior número de tarefas possível, das quais se incluem os *backups* diários, os procedimentos de rotina (por exemplo, limpeza de directórios), a actualização de programas e a monitorização dos processos do servidor e do espaço em disco.
4. **Definir ambientes de desenvolvimento e teste.** Deverá ser usado um ambiente para desenvolver e testar as aplicações antes de serem instaladas no sistema de produção, evitando assim resultados menos satisfatórios.
5. **Criar *stock* de componentes.** No caso de ocorrer uma falha de algum componente de *hardware*, embora não haja falha dos serviços prestados aos utilizadores, esse componente deverá ser substituído o mais rapidamente possível. Uma estratégia muito comum consiste em manter em *stock* alguns dos componentes de *hardware* susceptíveis de falhar. Deverão ser incluídos nesse *stock* os discos, as placas de rede e a cablagem diversa, entre outros componentes.

6. **Ajustar os contratos de manutenção de equipamentos críticos.** A implementação de arquitecturas robustas do ponto de vista de disponibilidade pode implicar uma redefinição dos contractos de manutenção dos equipamentos, com vista à diminuição do tempo médio de reparação e consequentemente do período de paragem. O *stock* de alguns componentes pode ser assegurado pelo próprio fornecedor.
7. **Definir o escalonamento de processos.** Deverão definir-se e escalonar-se tarefas para serem executadas pela equipa de gestão de sistemas em caso de falha. Pretende-se assim que os principais intervenientes conheçam os planos de contingência em vigor para situações de falha, como nos seguintes exemplos:
    - Está a ter lugar uma recuperação automática.
    - É requerida a intervenção do gestor do sistema após a ocorrência de uma falha.
    - É necessária a presença do suporte técnico para resolver um problema de *hardware*.
    - Ocorreu uma situação de desastre e é necessário iniciar os procedimentos de contingência.
8. **Planear situações de catástrofe.** É fundamental definir procedimentos claros sobre como a organização em geral, e os indivíduos em particular, devem lidar numa situação de desastre, como um incêndio ou um assalto. Estes procedimentos têm de ser claros e bem definidos, estipulando as funções a desempenhar por cada grupo (ou departamento) durante a ocorrência do desastre.
9. **Treinar.** Os intervenientes têm de ser treinados para pensar em termos de alta disponibilidade e *disaster recovery*. É igualmente importante, até mesmo fundamental, realizar testes periódicos que incluam todos os intervenientes. Estes testes deverão ser planeados e assentar na simulação de alguns cenários que possam ocorrer.
10. **Documentar:** Todos os detalhes da configuração do *hardware* e aplicações existentes, bem como todos os procedimentos utilizados devem ser documentados. Esta documentação deverá ser actualizada e revista periodicamente.

Um outro documento que deverá ser mantido e revisto com frequência é o registo de todas as operações excepcionais que acontecem no sistema bem como das acções correctivas que tiveram lugar para resolver o problema e o tempo de paragem.

Relativamente ao custo da solução, a primeira reflexão que deverá ser feita consiste em quantificar o prejuízo para o negócio devido a uma paragem não planeada da infra-estrutura (*hardware* e *software*) [29]. De seguida sumariam-se os pontos mais importantes para reflexão sobre a análise de custos:

- O custo da solução depende do nível de disponibilidade escolhida.
- O valor de uma solução de alta disponibilidade para a empresa está directamente relacionado com os custos de paragem do sistema.

- Quanto maior o custo de paragem, mais fácil se torna justificar o investimento gasto numa solução de alta disponibilidade.
- O custo da solução varia à medida que o nível estimado de disponibilidade se aproxima dos 100%.

3.5 SOLUÇÕES DE ALTA DISPONIBILIDADE

As tecnologias baseadas em clusters de servidores para implementar modelos de alta disponibilidade são disponibilizadas por praticamente todos os principais fabricantes de sistemas operativos, como por exemplo a HP [24] e a IBM [30]. Os conceitos que lhe estão associados são transversais às várias implementações, havendo no entanto características específicas de cada uma, como o preço, a escalabilidade e o desempenho das tarefas de retoma de recursos. Este facto leva a que seja possível a uma organização, num determinado momento e de acordo com o projecto em causa, optar pela solução mais adequada, com a curva de aprendizagem mais reduzida e com a melhor relação custo/benefício [21].

Relativamente ao sistema operativo AIX, a implementação de clusters de alta disponibilidade faz-se através da aplicação HACMP [30], descrita no Capítulo 4. Para as mesmas funções, a HP disponibiliza a aplicação ServiceGuard [24] para o sistema operativo HP-UX.

Os sistemas Linux têm igualmente investido no desenvolvimento de aplicações de *clustering* para alta disponibilidade, dos quais se destacam as soluções integradas no projecto Linux Virtual-Server (LVS) [31, 32], em particular as aplicações “heartbeat”, “ultramonkey”, “fake” e “mon”. Genericamente, um cluster LVS faz balanceamento de carga de vários serviços da rede por vários servidores, manipulando os pacotes à medida que são processados pela pilha protocolar TPC/IP [32]. Em [33] é possível encontrar alguns cenários práticos de clusters de alta disponibilidade com balanceamento de carga usando o “ultramonkey”. No Capítulo 5 detalha-se a configuração de clusters de alta disponibilidade usando o “heartbeat”, uma das aplicações incluídas no LVS. O cenário descrito permitiu aferir sobre a implementação de clusters de alta disponibilidade para serviços comuns de rede, recorrendo exclusivamente a soluções *open source*, com custos muito reduzidos de licenciamento, instalação e configuração.

O Common Address Redundancy Protocol (CARP) foi desenvolvido para o sistema operativo OpenBSD [34] e implementa simultaneamente as funções de alta disponibilidade e balanceamento de carga. É normalmente utilizado em conjunto com mecanismos de sincronismo de firewall (`pf` e `pfsync`) para criar um cluster de firewall redundantes e de alta disponibilidade [34-36].

Os clusters Beowulf são de desempenho escalável, assente em *hardware* do tipo COTS de simples utilização, com recurso a componentes aplicacionais OSS. São usados principalmente em aplicações que envolvam muitos recursos de processamento. Há vários projectos Beowulf, como o MAGI (República Checa), HIDRA (Espanha) e DeepFlow (Bélgica) [37].

O mercado do *open source,* mais concretamente as aplicações para Linux, é de tal forma apetecido que, também neste domínio, os grandes construtores disponibilizam as suas soluções proprietárias de alta disponibilidade para Linux. Por exemplo, a IBM e a HP disponibilizam respectivamente o HACMP e o ServiceGuard para o sistema operativo Linux [38].

A Microsoft disponibiliza vários produtos que implementam clusters de alta disponibilidade (Microsoft Cluster Server), destacando-se o "Network Load Balancing Services" (NLBS) para várias versões do sistema operativo [39]. O Windows2008Server (R2) denomina este tipo de produtos como "Failover Clustering", incluindo entre outras, soluções de virtualização de servidores através do Hyper-V [40].

Os protocolos de encaminhamento (por exemplo, RIP, OSPFS e EIGRP) também implementam mecanismos intrínsecos de redundância e de alta disponibilidade. Há outros protocolos que adoptam igualmente estratégias de *clustering* para assegurar essas funções no encaminhamento dos pacotes da rede. O protocolo Hot-Standby Redundancy Protocol (HSRP), proprietário da Cisco, implementa o conceito descrito. Além da redundância de caminhos, o HSRP pode igualmente ser configurado para balanceamento de carga, através da configuração de múltiplos grupos lógicos (Multiple HRSP). Neste caso, os vários *routers* pertencem aos mesmos grupos, mas desempenham um papel diferente (principal ou *standby*) [41]. O esforço de normalização do IETF traduziu-se na criação da especificação Virtual Router Redundancy Protocol (VRRP) [42], em tudo semelhante ao protocolo HSRP. O Global Load Balance Protocol (GLBP), igualmente proprietário da Cisco, é uma extensão do HRSP e está vocacionado para o balanceamento de carga através dos múltiplos *gateways* e *uplinks* redundantes instalados para *backup* dentro de uma rede IP [43].

3.6 NOÇÃO DE CLUSTER DE ALTA DISPONIBILIDADE

Um dos meios mais utilizados para implementar arquitecturas de alta disponibilidade é o recurso a clusters de servidores [24]. O exemplo descrito nesta secção é simples e normalmente implementado por vários sistemas operativos, como os apresentados nos Capítulos 4 e 5.

A primeira definição importante é a de cluster. No contexto de redes de computadores, um cluster pode genericamente ser definido como *"um conjunto de unidades de processamento agrupados em rede e partilhando os mesmos recursos de disco"***.** O conceito de cluster não é novo e, ao longo dos tempos têm aparecido configurações desse tipo associadas a alguns fabricantes. A Digital foi dos primeiros construtores a introduzir este conceito, através do sistema operativo VMS (Secção 2.2), utilizando para o efeito aplicações proprietárias que permitiam a partilha de uma área de disco por vários servidores. No entanto, estes modelos de cluster não implementam, na sua versão base, um modelo de alta disponibilidade.

Genericamente, há dois tipos de clusters: dedicados para obter **alto desempenho**, normalmente associado ao cálculo de operações complexas através de processamento paralelo; dedicados para obter **alta disponibilidade**, quando o servidor principal do cluster (ou um dos seus serviços) fica indisponível e um dos que estão configurados como *backup* assume as funções daquele.

Quando integrados num cluster, os servidores tomam a designação de "nó". Olhando para um nó isolado do cluster, a primeira acção no sentido de construir um cluster de alta disponibilidade é a identificação dos seus pontos críticos de falha. Um ponto crítico de falha é um componente (*hardware* ou *software*) que em caso de falha pode provocar a perda de acessibilidade aos serviços disponibilizados por um servidor. Normalmente, um componente que não tem qualquer elemento

redundante ou em *standby*, torna-se um desses pontos críticos de falha. Dependendo do cenário, a identificação dos pontos críticos de falha deve ser ponderada e ajustada ao orçamento disponível, sob pena de se definir uma infra-estrutura complexa e pesada em termos de administração, com um custo de implementação muito superior ao prejuízo que uma falha pode causar ao negócio.

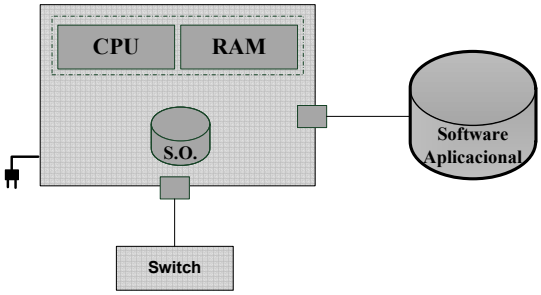


Figura 3-3 – Visão global dos componentes de um servidor.

Analisando a Figura 3-3, ilustrativa dos principais componentes de um nó, é possível identificar alguns dos possíveis pontos críticos de falha. A Tabela 3-2 apresenta algumas das falhas mais importantes que podem ocorrer neste sistema, o tipo de paragem associada e ainda a forma de solucionar o ponto de falha associado.

Componente	Consequência	Medidas
CPU	Falha do sistema até reparação do CPU.	Criação de um cluster, com pelo menos dois nós.
LAN lógica	Falha de ligação dos clientes.	Instalar duas placas de rede com endereços IP em sub-redes diferentes.
LAN física	Falha de ligação aos clientes.	Instalar duas placas de rede, ligadas a *switches* Ethernet diferentes.
Disco de S.O.	Falha de sistema até reparação do disco.	Usar *mirroring* do disco de sistema.
Disco com dados aplicacionais	Falha o acesso aos dados das aplicações. Eventual perda de dados durante a falha.	Usar *mirroring* dos discos da aplicação ou mecanismos de protecção, como RAID.
Alimentação eléctrica	Falha do sistema até haver novamente energia eléctrica.	Usar um sistema de alimentação alternativo do tipo UPS.
Sistema Operativo	O sistema pára, ficando eventualmente operacional no próximo *reboot*.	Usar sistemas operativos que implementem mecanismos de recuperação dos dados após falha.
Aplicações	Perda do serviço pelos utilizadores até resolver o problema existente com a aplicação.	Dotar as aplicações de mecanismos de arranque automático após uma falha e recuperação de eventuais erros após o arranque.
Erro humano	Falha do sistema até resolver o erro	Automatizar o mais possível os processos de operação e gestão das máquinas e aplicações, através da implementação de procedimentos sob a forma de *scripts*.

Tabela 3-2 – Principais pontos críticos de falha associados a alguns componentes do servidor.

A rede é provavelmente o componente mais importante, onde por vezes se torna difícil solucionar todos os eventuais pontos críticos de falha associados. Se seguirmos o trajecto de ligação de um cliente ao servidor podemos identificar vários pontos de falha, como sejam as placas de rede, os equipamentos passivos (cablagem) e os equipamentos activos (*switches* e *routers*). Ou seja, um problema na ligação dos utilizadores ao servidor pode envolver um ou mais destes equipamentos, sendo por isso necessário definir criteriosamente o *nível* de disponibilidade que se pretende atingir. Por exemplo, poderão ser integradas em cada servidor duas (ou três) placas de rede ligadas ao mesmo número de equipamentos activos distintos e instalados fisicamente em salas diferentes. No limite, poderão estar instalados em edifícios igualmente distintos.

Em suma, há duas questões importantes que devem ser realçadas na identificação dos pontos críticos de falha. A primeira refere-se à consciencialização de que pode não ser possível eliminá-los todos. São várias as razões e prendem-se normalmente com a quantidade de equipamentos (alguns muito específicos) que a rede possa ter. Em segundo lugar, deve igualmente ter-se a noção clara de que não é possível prever todas as situações. Por exemplo, não haverá nada a fazer se avariarem, simultaneamente, todas as interfaces de rede de um nó do cluster.

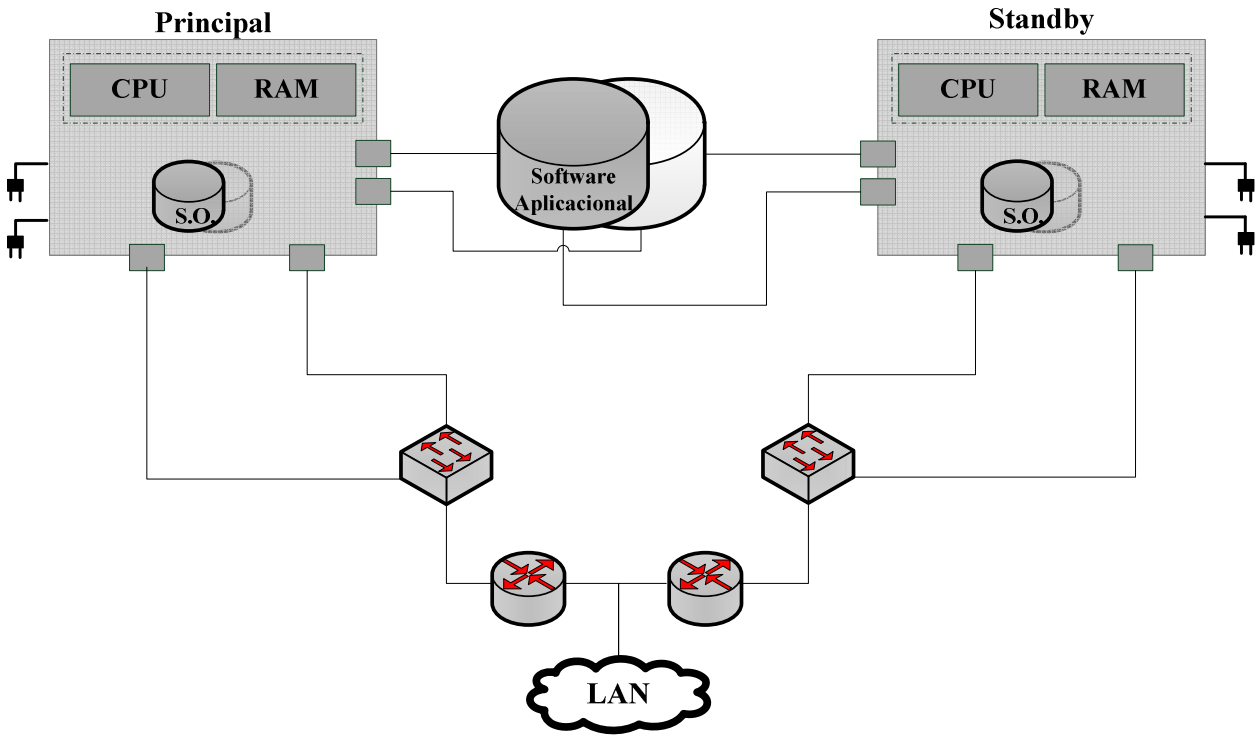


Figura 3-4 – Visão global de um cluster de alta disponibilidade.

A Figura 3-4 ilustra uma possível solução para um cluster com dois nós onde são eliminados os pontos críticos de falha identificados na Tabela 3-2. Assim:

- As interfaces físicas de rede estão duplicadas em cada nó, bem como os cabos de ligação.
- As redes LAN são diferentes e estão também ligadas a *switches* de Ethernet distintos.

- Os discos (internos e externos) dispõem de um sistema de *mirror*. São usados dois adaptadores distintos para ligar cada uma das cópias dos discos.
- Existem dois adaptadores de ligação à rede de abastecimento de energia eléctrica.

À semelhança dos componentes de *hardware* descritos acima, também os programas podem originar paragens não planeadas do sistema. Estas paragens são normalmente originadas por dois tipos de falhas: análise de requisitos pobre que se traduz na não concretização de todas as situações possíveis; desenvolvimento defeituoso, permitindo conflitos de acesso a memória e outros erros que provocam normalmente a paragem anormal do programa.

Por esse facto, as aplicações devem garantir sempre mecanismos de consistência dos dados após uma falha. Assim, durante a fase de configuração do modelo de alta disponibilidade deve ter-se em conta a construção de procedimentos de monitorização da aplicação e da sua activação em caso de falha. As aplicações devem ainda ter algumas características desejáveis para que possam ser facilmente integradas num sistema deste tipo:

- Capacidade de transferir o controlo da aplicação para outro nó no caso de uma falha, sem perda dos dados.
- Capacidade de recomeçar os seus serviços, garantindo consistência e coerência dos dados após uma falha.
- Permitir a ligação dos clientes a vários sistemas remotos e não apenas a um.
- Possibilidade de se efectuar a monitorização dos seus serviços, a fim de se saber se está ou não activa.
- Desenvolver procedimentos bem documentados de paragem e arranque da aplicação.
- Desenvolver procedimentos bem documentados de *backup* e recuperação dos dados.
- Definir e documentar mecanismos de actualização da aplicação.

Outro aspecto importante no desenvolvimento de clusters, onde se incluem os dedicados a alta disponibilidade, é a necessidade de sincronizar os relógios dos nós. Esta importância deve-se ao facto de haver várias operações do cluster, onde se inclui o processamento dos eventos, que se baseiam na comparação da data e hora actual dos servidores. A não sincronização correcta dos eventos poderá levar à inviabilização de algumas operações no cluster. Há várias formas de implementar mecanismos de sincronismo temporal dos relógios dos nós do cluster, como por exemplo, usando o protocolo Network Time Protocol (NTP) [44, 45], ou ainda tirando partida de programas específicos para determinadas arquitecturas, como na arquitectura SP (Secção 2.4) [15].

3.6.1 GRUPO DE RECURSOS

Um conceito importante em sistemas de alta disponibilidade é o de grupo de recursos (*resources group*). Um grupo de recursos define-se como o conjunto de componentes que são transferidos do nó principal do cluster para um de *standby* em caso de uma paragem (programada ou não) daquele. Nesse caso, é executado um procedimento que "transfere" os componentes do

grupo de recursos de um nó para o outro. Podem fazer parte do grupo de recursos, entre outros, partições lógicas de discos, *filesystems*, serviços de rede e aplicações.

No grupo de recursos devem igualmente definir-se as relações entre os nós, ou seja, qual é o principal e qual o de *standby*. O nó principal corresponde ao que ficará activo por omissão no arranque do cluster. O de *standby* corresponde ao que entra em funcionamento quando ocorre um evento no nó principal que origine a retoma do grupo de recursos do cluster. Nalguns protocolos ou aplicações de alta disponibilidade, o nó principal é também designado de activo ou primário. Quanto ao nó de *standby*, aparece também referido como sendo de *backup* ou secundário. Ao longo do relatório serão usados os termos "principal" e "*standby*".

A forma como os nós retomam os recursos do cluster em caso de falha pode ser de três tipos: em cascata, em rotação ou de acesso concorrente. Em ambos os tipos existe sempre um nó que desempenha, em cada instante, o papel de principal e outro que desempenha a função de *standby*. Quando o método a utilizar é em cascata, o nó principal assegura os serviços inicialmente. Se houver uma falha desse nó, os recursos são transferidos automaticamente para o de *standby*. Quando o sistema que inicialmente falhou voltar a ficar disponível e integrado no cluster, é ele que toma novamente o controlo dos recursos. No método de rotação, uma falha do nó principal implica a retoma de serviços e promove o de *standby* a principal. O restabelecimento posterior do actual nó de *standby* não implica a retoma de recursos em sentido inverso. Estes dois tipos de retoma dos grupos de recursos são ilustrados na Figura 3-5, onde se pode constatar o estado final do cluster após uma falha do nó principal e a sua posterior operacionalidade.

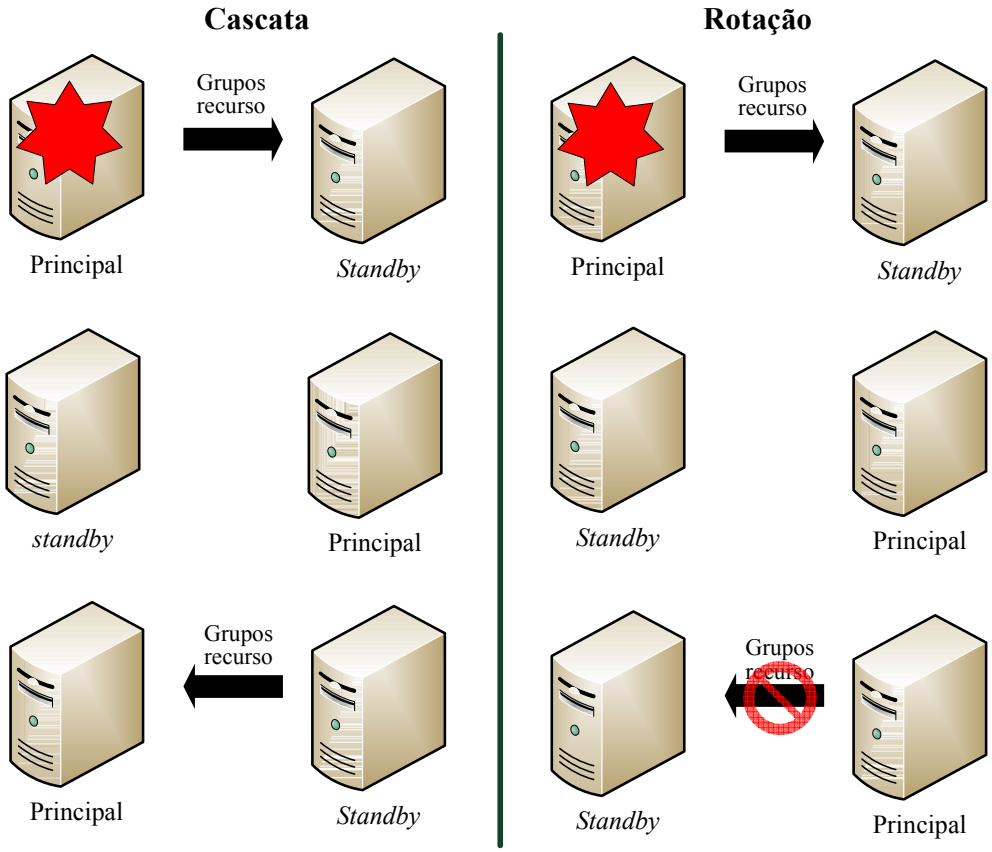


Figura 3-5 – Estratégia de retoma de serviços em cascata e rotação.

As estratégias de retoma de serviços em cascata e rotação são implementadas recorrendo a uma configuração de *standby*. Ou seja, há um nó que se encontra em *standby* e que assume o controlo do cluster sempre que o nó principal falha. Neste caso, a transferência dos recursos é efectuada de cada vez num só sentido e o acesso aos discos externos é não concorrente. A utilização de uma estratégia em cascata ou rotação tem essencialmente a ver com a configuração de *hardware* dos nós. Por exemplo, se a configuração do nó principal for substancialmente superior à do de *standby*, é aconselhável utilizar uma configuração em cascata. Nesse caso, o nó de *standby* apenas assumirá o controlo do cluster durante o período de tempo em que nó de maior configuração (o principal, por omissão) volte a ficar operacional.

A última estratégia apresentada, de acesso concorrente, está normalmente associada a uma configuração de *mutual takeover*, sendo a transferência realizada em ambos os sentidos. Ou seja, qualquer um dos nós pode retomar os serviços do outro em caso de falha. Este facto leva a que se devam implementar mecanismos de protecção para que ambos os nós do cluster acedam simultaneamente aos discos externos. Neste tipo de configuração, ambos os nós desempenham o papel de "principal" alocando primariamente um grupo de recursos. Ou seja, cada nó assume o papel de *standby* do grupo de recursos do outro nó. Quando um dos nós falha, o outro retoma os componentes configurados nesse grupo de recursos, podendo assegurar simultaneamente, durante um determinado período de tempo, o controlo de dois grupos de recurso. Assim, neste tipo de configuração, os nós têm de ter uma configuração de *hardware* que permita retomar em simultâneo dois grupos de recurso.

Através de uma configuração de *standby* é possível obter o mesmo resultado, usando três nós: dois principais para cada grupo de recursos e um terceiro de *standby* para ambos. Ou seja, o nó de *standby* assume esse papel para ambos os grupos de recursos detidos pelos nós principais. Se houver uma retoma dupla, por exemplo devido à paragem simultânea dos nós principais, o nó de *standby* assumirá o papel de nó principal para ambos os grupos de recursos.

Em suma, a utilização de clusters para obter continuidade dos serviços em caso de falha pode, em certa medida, complementar-se com a implementação de soluções de balanceamento de carga. Por exemplo, o uso de configurações de *mutual takeover* permite, por um lado a retoma dos serviços dos nós primários de cada cluster e, por outro, a distribuição de pedidos de duas aplicações distintas por cada nó do cluster.

3.6.2 RETOMA DE ENDEREÇOS

A Figura 3-6 ilustra a operação de retoma do endereço IP, normalmente designada de *IP takeover*, que consiste em associar o endereço IP de serviço à interface Ethernet do nó de *standby*. Além do endereço IP para acesso interno, a interface passa a ter mais um: o endereço IP de serviço. A partir desse momento o nó de *standby*, entretanto promovido a principal, está apto para processar o tráfego e responder a pedidos ARP efectuados ao endereço de serviço. No entanto, ao iniciar a sua função não é garantido que o tráfego para este endereço seja imediatamente recebido pelo servidor, já que os clientes do cluster (utilizadores) possuem o endereço IP virtual associado ao endereço MAC do servidor que falhou (servidor principal) e não o seu. Logo, só depois de o cliente efectuar novamente um pedido ARP é que o servidor envia a resposta com o seu endereço MAC. Sendo assim, o cliente actualiza a entrada da sua tabela de ARP relativa ao endereço IP

virtual. Ou seja, apesar da possibilidade de o servidor principal estar inacessível, pode ainda responder a pedidos ARP efectuados ao endereço IP virtual com o seu endereço MAC. Caso isto aconteça, de cada vez que um utilizador envia um pedido ARP, alguns pacotes podem ser enviados para o servidor principal, mesmo que este se encontre inoperacional.

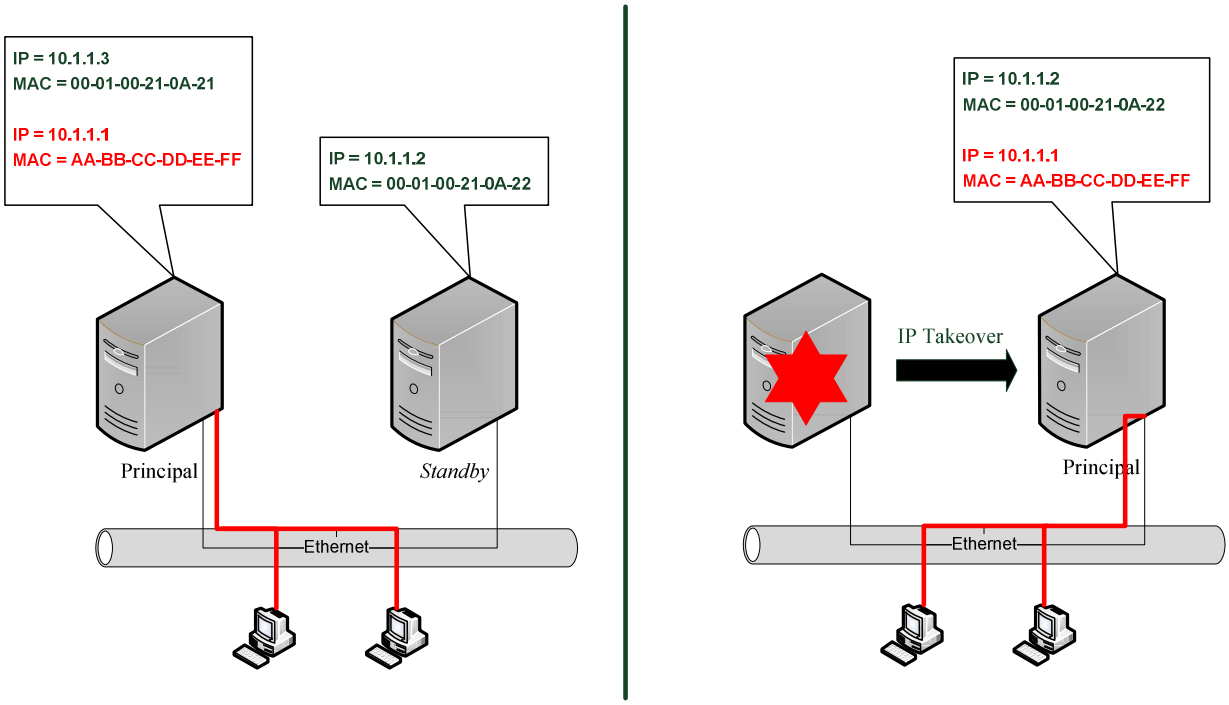


Figura 3-6 – Reconfiguração dos endereços IP e MAC no processo de retoma de recursos.

Paralelamente, também o endereço MAC de serviço, configurado normalmente com um endereço com pouca probabilidade de ser utilizado pelos construtores (por exemplo, `AA-BB-CC-DD-EE-FF`), é retomado para o nó de *standby*. Quando ocorre um *takeover* de endereço IP, os clientes são confrontados com um endereço MAC diferente para o mesmo endereço IP. A RFC 826 [46] especifica que todas as implementações IP devem actualizar as tabelas de ARP caso ocorra um pedido ARP para o qual já exista um endereço IP associado. Assim, é possível actualizar a tabela de ARP depois de ocorrer o *takeover* de endereço IP, realizando para tal uma operação de `ping` aos clientes.

3.7 PLANEAMENTO

A metodologia usada na implementação de um cluster de alta disponibilidade está ilustrada na Figura 3-7. Esta metodologia foi adoptada com sucesso na implementação de vários cenários de alta disponibilidade usando clusters Unix, como os exemplos expostos nos Capítulos 4, 5 e 6.

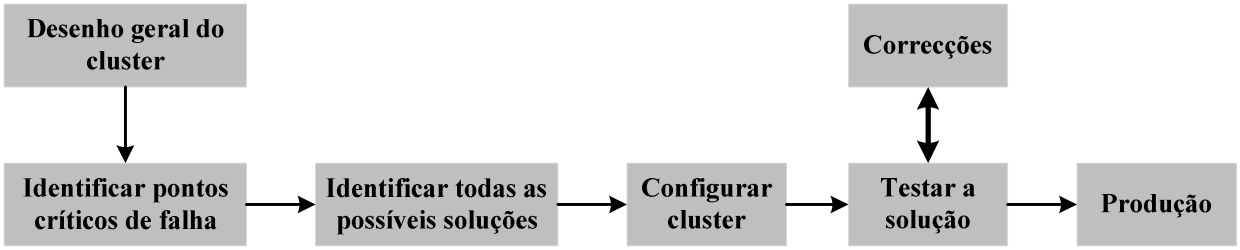


Figura 3-7 – Metodologia adoptada na implementação de um cluster de alta disponibilidade.

De uma forma geral, o planeamento deve incluir as seguintes fases:

1. Identificar os blocos principais do cluster, nomeadamente reunir toda a documentação sobre as aplicações e os recursos que deverão ser retomados. Deverão ainda analisar-se as estratégias de *takeover* e acesso aos discos a implementar. É necessário reservar uma rede de endereços IP privados e alocá-los genericamente a cada uma das interfaces de rede. Neste primeiro passo o objectivo é apresentar uma visão global do cluster, onde se identifiquem os principais blocos e as principais iterações. Deverá igualmente servir para iniciar a documentação técnica do projecto.
2. Identificar os pontos críticos de falha da infra-estrutura que se pretende dotar de mecanismos de alta disponibilidade. Embora seja difícil eliminar todos os pontos críticos, bem como prever todas as situações possíveis que possam ocorrer, é no entanto possível identificar, dentro do bom senso, os que são mais críticos e que têm um maior leque de soluções que possam ser adoptadas.
3. Dos que foram extraídos do ponto anterior, avaliar qual a melhor solução para cada um.
4. Configurar o cluster, que inclui principalmente a infra-estrutura de rede, os servidores, os discos, a aplicação de gestão do cluster e as aplicações de negócio.
5. Testar a solução, definindo baterias de testes que contemplem o maior número de cenários possível.
6. Realizar ajustes à configuração de acordo com os testes realizados. Após obter uma configuração estável do cluster, disponibilizá-la no ambiente de produção.

Sobre a infra-estrutura de rede deverão identificar-se as diversas opções relativas à configuração de redes IP, nomeadamente se serão usadas ligações a bastidores de comunicação diferentes, a *switches* ou a *routers* distintos [47]. Deverá igualmente verificar-se a necessidade de implementar uma rede alternativa entre os nós do cluster. A solução recai normalmente na utilização de uma ligação série RS232 como forma de eliminar o ponto crítico de falha relativo ao protocolo Ethernet.

Relativamente à necessidade de configurar, da melhor forma possível, os recursos de disco disponíveis para serem partilhados pelo cluster, devem definir-se algumas normas e regras relativas à utilização dos discos físicos do cluster:

- Definir nomes sugestivos para as partições lógicas dos discos.

- Definir políticas para a criação de *filesystems* e sua localização em termos de acesso aos dados.
- Implementar mecanismos de *mirroring* e *stripping* de discos. Os discos relativos à primeira cópia devem estar ligados fisicamente a um adaptador diferente dos discos da segunda cópia. Esta estratégia deverá ser igualmente seguida para o disco do sistema operativo.

As aplicações críticas deverão estar todas associadas a um grupo de recursos do cluster. Desta forma, todas as aplicações serão retomadas aquando de um evento de falha, simplificando a gestão do cluster a vários níveis:

- Os restantes componentes do grupo de recursos são os mesmos para todas as aplicações.
- A retoma de recursos é mais rápida já que apenas é lido e executado um único grupo de recursos.
- A existência de apenas um *script* de arranque e paragem das aplicações facilita a tarefa de gestão do cluster.

Em cenários reais recorrendo a configurações de *standby*, como o que é descrito no Capítulo 4, é muito comum utilizar o servidor de *backup* para tarefas auxiliares, não críticas, durante o período em que está inactivo e não tem o controlo do grupo de recursos. Por exemplo, é muito frequente que esse servidor seja utilizado como servidor de testes. No entanto, estas aplicações não deverão constar do grupo de aplicações definidas para retoma automática em caso de falha. Estas aplicações são apenas acedidas localmente, através da interface local do servidor e, em caso de necessidade de retoma de recursos, são imediatamente desactivadas e os utilizadores desligados, dando prioridade ao processo de retoma automática das aplicações realmente críticas.

As aplicações de *clustering* têm normalmente potencialidades ao nível de processamento de eventos que devem ser aproveitadas. É normalmente possível associar *scripts* a cada tipo de evento, como por exemplo:

- Definição de *scripts* executados imediatamente antes (“pre”) ou depois (“pos”) do processamento de um evento no cluster.
- Notificação automática sobre eventos não directamente controlados pela aplicação de clustering, como por exemplo o estado das aplicações e de sistemas gestores de bases de dados.

No caso de estudo descrito no Capítulo 4 tirou-se igualmente partido destas funcionalidades disponibilizadas pela aplicação HACMP para AIX.

4 CASO DE ESTUDO 1 - ALTA DISPONIBILIDADE EM AIX

Os conceitos descritos no Capítulo 3 são transversais a várias arquitecturas de computadores e sistemas operativos que possibilitem a implementação de clusters de alta de disponibilidade. Este capítulo começa por descrever, em linhas gerais, o projecto Tradutor de EDI (TEDI) e a aplicação de *clustering* HACMP para sistemas AIX, em arquitecturas IBM RISC/6000. De seguida, detalham-se as acções de planeamento e configuração levadas a cabo durante a implementação do HACMP no projecto TEDI, o primeiro no grupo Sonae a ser configurado com esta tecnologia em 1997 [48]. Por fim, descrevem-se os testes realizados e os resultados obtidos.

4.1 PROJECTO TEDI

A competitividade que já se fazia sentir no negócio do retalho veio demonstrar às organizações, nomeadamente à Sonae Distribuição, a necessidade de uma gestão mais adequada do fluxo de produtos ao longo da cadeia de fornecimento das lojas. A transferência electrónica de documentos, *Electronic Data Interchange* (EDI), é uma ferramenta essencial para a troca de documentos (notas de encomenda, facturas, etc.) por via electrónica, traduzindo-se num consequente aumento de eficácia e produtividade. Assim, o esforço de implementação é dirigido de maneira a que o fornecedor cumpra os requisitos do cliente, de forma segura e consistente, através de uma cadeia de fornecimento que tem em conta o binómio custo-eficácia. Juntamente com o facto de cada empresa ser simultaneamente cliente e fornecedor, tanto a montante como a jusante da cadeia de fornecimento, faz do EDI um meio poderoso para o aperfeiçoamento da gestão nas empresas de retalho [49, 50].

Os dados relativos a vários processos de negócio, como as encomendas, devoluções ou facturação, existem em documentos em papel que são obtidos a partir de aplicações informáticas, nomeadamente de gestão de produção, *stocks* ou inventário. Do lado do fornecedor, estes dados são processados e posteriormente impressos num formato padronizado, sendo de seguida enviados pelo correio ou por fax para o cliente. Este, por sua vez, volta a introduzir toda a informação no seu sistema de informação para novo processamento, completando assim o ciclo de tratamento da informação ao longo da cadeia.

O tradutor de EDI (TEDI) surgiu devido à necessidade de optimizar o fluxo de informação entre os clientes e fornecedores ao longo da cadeia de abastecimento sem recurso a papel, utilizando apenas a transferência dos dados de forma electrónica e automática, com mínima intervenção humana. Os dados são enviados de forma estruturada, através de mensagens normalizadas e previamente acordadas entre o cliente e o fornecedor. A funcionalidade básica de um tradutor EDI consiste assim em converter uma mensagem normalizada para o formato esperado no sistema de informação da organização receptora (cliente ou fornecedor) e inversamente no caso das mensagens recebidas. Estas aplicações devem ainda assegurar o envio automático de mensagens para clientes que não possuam um sistema de informação assente em aplicações informáticas.

A Figura 4-1 compara o modelo de troca de documentos tradicional com o electrónico que recorre ao uso de um tradutor de EDI. Na cadeia de distribuição existem normalmente alguns fornecedores que não possuem um mecanismo automático de integração electrónica de dados no seu sistema de informação. Para contemplar também estes fornecedores, os tradutores de EDI possuem normalmente mecanismos automáticos de envio destes dados por fax.

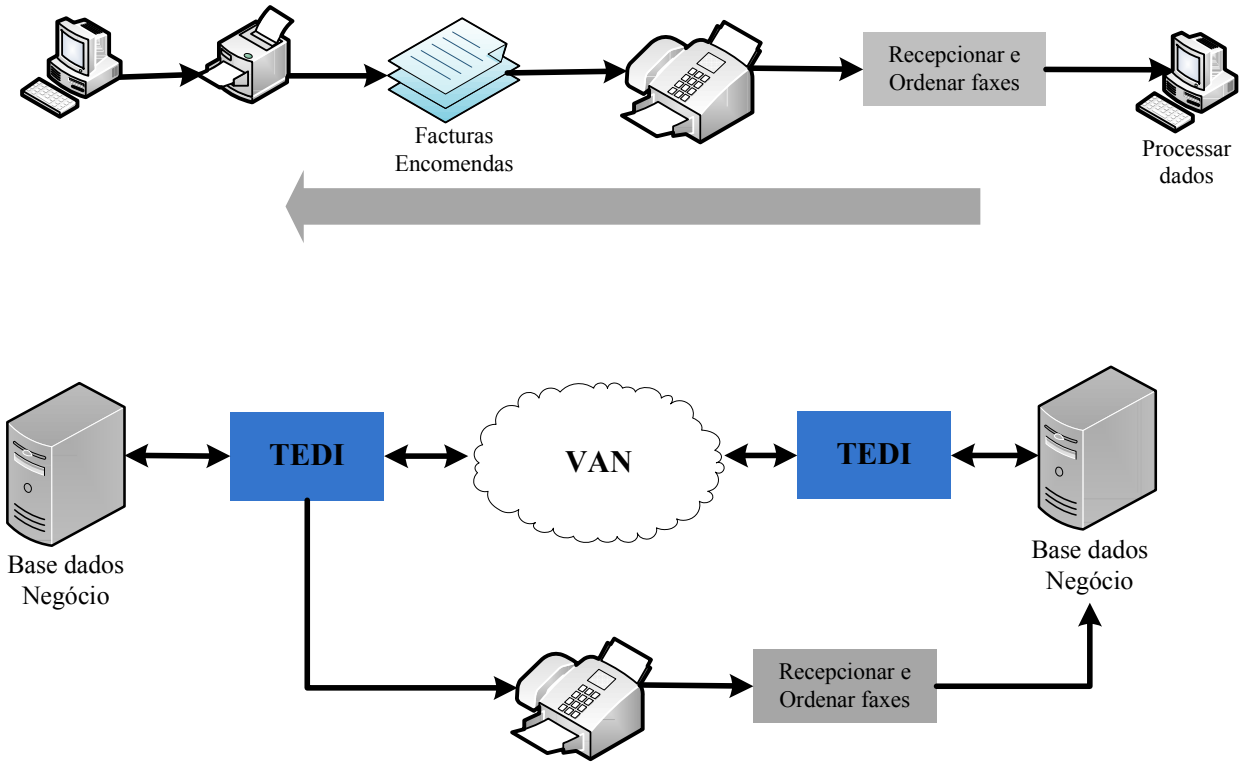


Figura 4-1 – Comparação entre o modelo tradicional e o EDI para troca de informação.

A relação entre os custos e benefícios da implementação de um tradutor EDI podem traduzir-se em [50, 51]:

- Custos de estratégia, desenvolvimento, formação e implementação da aplicação.
- Poupança nos custos administrativos e de processamento.
- Benefícios proporcionados por um circuito comercial mais rápido.
- Benefícios estratégicos e de competitividade.

O processo de troca de mensagens entre duas empresas utilizando o EDI é assegurado, normalmente, por uma rede de valor acrescentado (Value Added Network - VAN). A VAN tem por finalidade gerir as mensagens dirigidas a cada uma das empresas com a qual o cliente (ou fornecedor) tem relações comerciais. Assim, quando uma empresa envia um documento EDI para outra, fá-lo primeiro para a VAN. Esta encarrega-se de "ler" o destinatário da mensagem e colocar a mensagem na "caixa de correio" respectiva. O destinatário da mensagem, através de um processo automático, ou não, vai recolher electronicamente todas as mensagens que tem na sua "caixa de correio". Internamente deverá implementar processos de normalização e integração das mensagens recebidas para o seu sistema de informação.

Em empresas onde o número de mensagens enviadas é muito elevado e existe uma rede de distribuição a trabalhar com base nas encomendas recebidas via EDI, é necessário questionar se o sistema que suporta o tradutor de EDI e toda a infra-estrutura de troca de mensagens pode estar indisponível. E se sim, qual a duração máxima tolerada. Tal situação foi considerada inaceitável por parte da gestão de negócio de retalho da Sonae Distribuição, tendo-se definido este projecto como crítico e, por isso, susceptível de serem definidos mecanismos de alta disponibilidade e tolerância a falhas. As razões na altura eram óbvias e continuam actuais nos dias de hoje: a paragem no envio automático de encomendas aos fornecedores origina falhas no planeamento de inventário nos centros de distribuição e, consequentemente, nas entregas de mercadoria às lojas, originado quebras de *stock* de produtos e, a consequência mais dramática, a quebra nas vendas. Assim, este era seguramente um projecto em que o investimento na infra-estrutura de alta disponibilidade teria impacto a médio prazo nas receitas geradas pelas lojas [51].

A realidade à data do desenvolvimento deste projecto era a não existência de um sistema de EDI na maioria das empresas, recorrendo-se normalmente ao envio da documentação por fax ou correio. Por exemplo em 1998, na fase inicial de utilização do TEDI na Sonae Distribuição, a taxa de envio de mensagens EDI cifrava-se nos 30%, embora apenas quatro fornecedores recebessem relatórios de inventário por EDI. Dois anos mais tarde, o volume de encomendas enviadas electronicamente subiu para os 50%, abrangendo apenas 10% dos fornecedores envolvidos. Desses, apenas dois terços integravam o EDI com o sistema integrado de gestão da empresa [52].

Genericamente, um tradutor de EDI contempla um conjunto de componentes aplicacionais que têm como objectivo [53, 54]:

- Converter os dados dos vários tipos de mensagens recebidas para um formato normalizado.
- Assegurar o envio e recepção de mensagens através de fax, para os clientes cujo sistema de informação não está preparado para receber mensagens EDI. No projecto em causa foi utilizada uma unidade de fax com seis linhas configuradas para envio.
- Assegurar o envio de mensagens EDI para as empresas que se encontrem preparadas para o efeito, recorrendo a uma rede de comunicações de valor acrescentado (VAN).
- Registar todas as operações de envio e recepção, quer das mensagens EDI quer dos faxes.

O projecto Tradutor EDI (TEDI) na Sonae Distribuição consistiu na implementação de uma infra-estrutura aplicacional e de rede, cuja principal motivação consistiu na implementação das funcionalidades de EDI na cadeia de abastecimento das lojas de retalho do grupo, desde o fornecedor de produtos (que podia ser o produtor) até à loja (por exemplo, Modelo e Continente), passando pelo centro de distribuição. Pelo seu grau de criticidade para o negócio, este projecto acolheu a primeira implementação de uma solução de HACMP para sistemas AIX, realizada no universo de aplicações do grupo Sonae. Decorreu durante o ano de 1997 e constituiu a base para futuras implementações que se realizaram posteriormente em aplicações críticas. Além da implementação prática de uma solução com estas características num projecto crítico para a Sonae, pretendeu-se ainda definir um conjunto de boas práticas que serviriam de base para futuras implementações. Tal foi conseguido, não só pela identificação dos problemas durante o planeamento, configuração e testes, mas também pela documentação de todas as acções levadas a cabo, algumas delas descritas nesta secção.

4.2 A APLICAÇÃO HACMP PARA IBM AIX

O HACMP, cuja sigla deriva de "High Availability Cluster Multi-Processing", é um pacote aplicacional comercializado e distribuído pela IBM, para a implementação de clusters de alta disponibilidade em sistemas RISC/6000, com sistema operativo AIX [55, 56]. Esta aplicação foi desenvolvida originalmente pela empresa Clam[4], tendo passado o seu desenvolvimento para a IBM em Agosto de 2006. O HACMP tem duas componentes principais; *High Availability (HA)* e *Cluster Multi-Processing (CMP).* A componente de "HA" refere-se à implementação do conceito de alta disponibilidade e a componente "CMP" refere-se ao facto de ser possível a execução de várias aplicações no mesmo nó, com acesso partilhado ou concorrente aos dados. O HACMP suporta os dois tipos de configuração de grupos de recursos: *standby* e *mutual takeover.*

A Figura 4-2 enquadra o HACMP na estrutura de *software* existente num sistema Unix. Trata-se de uma camada aplicacional que interliga as aplicações do negócio e o sistema operativo, composta entre outros pela gestão de rede e de discos. A abrangência espacial deste tipo de aplicações é normalmente local, estando os vários nós do cluster próximos uns dos outros. É muito usual, como aconteceu no projecto TEDI descrito de seguida, ter um cluster com dois nós, fisicamente instalados no mesmo edifício mas em divisões distintas.

Figura 4-2 – Enquadramento das aplicações de clustering de alta disponibilidade.

O HAGeo [57-59], também desenvolvido e comercializado pela IBM, é uma variação do HACMP que permite que os nós do cluster distem entre si vários quilómetros, incrementando assim o nível de disponibilidade da solução em situações de destruição completa (ou parcial) das instalações. As mensagens de validação do estado de cada um dos nós são enviadas através de uma ligação normalmente privada e dedicada exclusivamente para o cluster. Através do HAGeo é possível implementar mecanismos de *mirroring* remoto dos discos, através da aplicação GeoRM [57]. O nó principal do cluster, após a escrita nos discos locais, envia uma mensagem ao nó de *standby* referindo que vai enviar dados para serem escritos nos discos de *mirroring* remotos. O nó de *standby* escreve os dados e envia uma resposta com o sucesso ou insucesso da operação.

O HACMP é composto pelos seguintes processos principais:

[4] Actualmente designada LakeViewTech (http://lakeviewtech.net/)

- **`Cluster Manager:`** processo que se encontra permanentemente activo em cada nó do cluster. Tem como principal função a monitorização dos nós sobre eventuais falhas. Para o efeito, são enviadas mensagens UDP de controlo, para os nós a partir das interfaces de rede configuradas. É ainda responsável pela comunicação com os processos homólogos nos restantes nós e por iniciar os procedimentos associados a eventos.
- **`Cluster SMUX Peer:`** processo responsável por informar os clientes sobre eventuais alterações à topologia do cluster.
- **`Cluster Information:`** processo que actualiza a informação proveniente do processo `Cluster SMUX Peer` e disponibiliza-a aos clientes gráficos de monitorização.
- **`Cluster Lock Manager:`** este processo é apenas utilizado num modelo de acesso concorrente, cujo objectivo é garantir a integridade dos dados acedidos desta forma.

O HACMP responde a falhas de um componente do cluster (por exemplo, interface de rede) através de eventos detectados pelo processo `Cluster Manager`. Em cada nó, um evento pode ser meramente informativo ou, por outro lado, realizar um conjunto de acções pré-definidas. Os eventos onde são tomadas acções pré-definidas surgem normalmente associados a falhas em interfaces de rede, bem como ao próprio nó, podendo ocorrer tanto no nó principal como no de *standby*. Quando um nó não recebe resposta às mensagens de controlo UDP enviadas a partir de uma das interfaces de rede, o HACMP despoleta um evento no cluster para essa interface. Quando não recebe resposta de nenhuma interface de rede, é gerado um evento associado ao próprio nó. Em ambos os casos é executado um *script* com um conjunto de acções relacionadas com o evento em causa. Essas acções são executadas apenas num nó ou em ambos, conforme o tipo de evento. Os *scripts* podem ainda ser do tipo "pre" ou "pos" qualquer evento, sendo executados respectivamente antes ou depois da ocorrência. Por exemplo, relativamente ao evento "node_down", é possível definir um outro "pre_node_down" e associar um *script* que execute as operações relacionadas com o tratamento prévio do evento. Da mesma forma, é possível executar um *script* associado ao final do evento "node_down", executado após o evento "node_down_complete". No Anexo F são resumidos alguns dos principais eventos definidos no HACMP.

O HACMP dispõe ainda de vários utilitários gráficos de apoio à configuração e monitorização do cluster, dos quais se destacam o `xclconfig, xclstat, xhacmpm` e `cldiag`**.** O uso destas aplicações permite desenvolver processos enquadrados na rotina de monitorização e operação da rede, acelerando a activação de procedimentos de contingência em caso de falha nalgum dos componentes do cluster.

A paragem do HACMP é normalmente realizada durante o processo de paragem de um nó, dispondo de três opções possíveis:

- **forced.** A paragem dos processos é imediata e não há lugar a retoma das aplicações por parte do outro nó. Esta opção deverá ser utilizada apenas quando o cluster está num estado instável e não há grandes possibilidades de recuperação sem parar os processos de HACMP.

- **graceful.** Nesta opção o nó liberta os recursos do cluster que possui no momento, não havendo lugar a retoma dos recursos por parte do nó de *standby*.
- **graceful with *takeover*.** Esta opção deve ser utilizada quando se pretende uma paragem normal dos processos de HACMP com retoma dos serviços por parte do nó de *standby*.

O arranque dos processos de HACMP é realizado automaticamente no arranque de qualquer nó ou quando há uma retoma dos recursos por parte do nó de *standby*. O Anexo G ilustra a sequência de operações que ocorre na paragem do HACMP num nó do cluster com a opção "graceful with takeover".

No HACMP as redes são classificadas de acordo com a sua função no cluster. Assim, logicamente as redes podem ser públicas ou privadas. Uma rede pública é constituída pelas interfaces externas, acessíveis a partir do exterior do cluster. Define-se uma rede como privada quando apenas é acessível internamente, através dos nós. Sobre este assunto, devem considerar-se os seguintes aspectos:

- Numa configuração de acesso concorrente com *mutual takeover*, utilizando o `Cluster Lock Manager`, deverá ser usada, preferencialmente, uma rede privada para o efeito.
- As redes privadas servem para uso restrito dos nós. No caso em estudo, a rede constituída pelo *switch* do SP e pelas interfaces de ligação interna entre os nós são, por omissão, redes privadas e, portanto, sem acesso do exterior. A rede IP constituída pelas interfaces internas do SP tem endereços IP privados, apenas acessíveis no interior do cluster SP.
- A ligação série estabelecida entre os nós (RS232) é sempre uma rede privada e só é visível pelos nós.
- As interfaces de rede Ethernet com acesso a partir do exterior do cluster constituem uma rede pública.

Dependendo do tipo de aplicações e da forma como irão ser acedidas, pode haver interesse em definir apenas redes públicas, somente privadas ou ambas. No caso concreto do projecto TEDI foram definidas duas redes, uma pública e outra privada. A rede privada era constituída pela ligação ao *switch* e à Ethernet interna do SP, juntamente com a ligação RS232 entre os nós. Quanto à rede pública, integrou duas interfaces Ethernet em cada um dos nós, conforme se detalha na Secção 4.3.

Do ponto de vista do HACMP, as interfaces de rede podem ser de três tipos distintos:

- **Service:** Esta interface configura a ligação do nó à rede Ethernet acedida externamente, responsável por processar o tráfego enviado pelos clientes para o cluster.
- **Boot:** Identifica o endereço configurado no arranque do cluster ou após uma paragem do nó primário e consequente retoma para o nó de *standby*. Deve existir um endereço de *boot* em cada interface de serviço, por cada nó que possa efectuar a retoma de recursos. Quando ocorre uma falha no nó principal, o de *standby* assume os recursos do cluster através de uma interface de serviço, havendo lugar a um *takeover* do endereço IP de serviço do nó primário para o de *standby*. Contundo, não há *takeover* do endereço IP de *boot*, mantendo-se no nó primário. Quando se processar a reintegração do nó primário novamente no

cluster, o endereço de *boot* é utilizado para arrancar com os serviços do cluster. Após a desconfiguração do endereço de serviço pelo nó de *standby*, o principal efectuará a sua reconfiguração na sua interface de serviço. Este endereço IP não está associado a uma interface diferente. Na realidade, a interface de serviço do nó primário é configurada, conforme o estado do cluster, com um endereço de serviço ou de *boot*. Se a estratégia usada for de *mutual takeover*, cada nó terá configurado um endereço de *boot*.

- ***Standby:*** Trata-se de uma interface física de *backup* à interface configurada com o endereço de serviço. Assim, se a interface de serviço do nó falhar, o processo `Cluster Manager`, através do evento "swap_adapter" (Anexo F), configura o endereço IP na interface de *standby*, não implicando perda de acessibilidade para os utilizadores. Um nó pode não ter nenhuma interface de *standby* configurada no HACMP, ou mais do que uma até ao máximo de sete por cada tipo de rede utilizada. No projecto TEDI foi configurada uma interface de *standby* para a rede Ethernet externa em cada nó do cluster.

Relativamente à configuração das interfaces de rede no HACMP:

- As redes lógicas definidas atrás devem ter um nome sugestivo.
- Todos os endereços IP associados às interfaces de rede devem ser resolvidos pelo serviço de DNS e/ou localmente através da actualização do ficheiro `/etc/hosts`.
- Cada rede configurada no HACMP deverá estar associada a um dispositivo do sistema operativo. No AIX os dispositivos utilizados foram os seguintes: Ethernet (`en`), *switch* do SP (`css`) e ligação RS232 (`serial`).

4.3 ARQUITECTURA ADOPTADA

A Figura 4-3 ilustra a solução final de alta disponibilidade proposta para o projecto TEDI [48]. Em [60, 61] podem igualmente encontrar-se vários exemplos de implementação de HACMP em clusters SP.

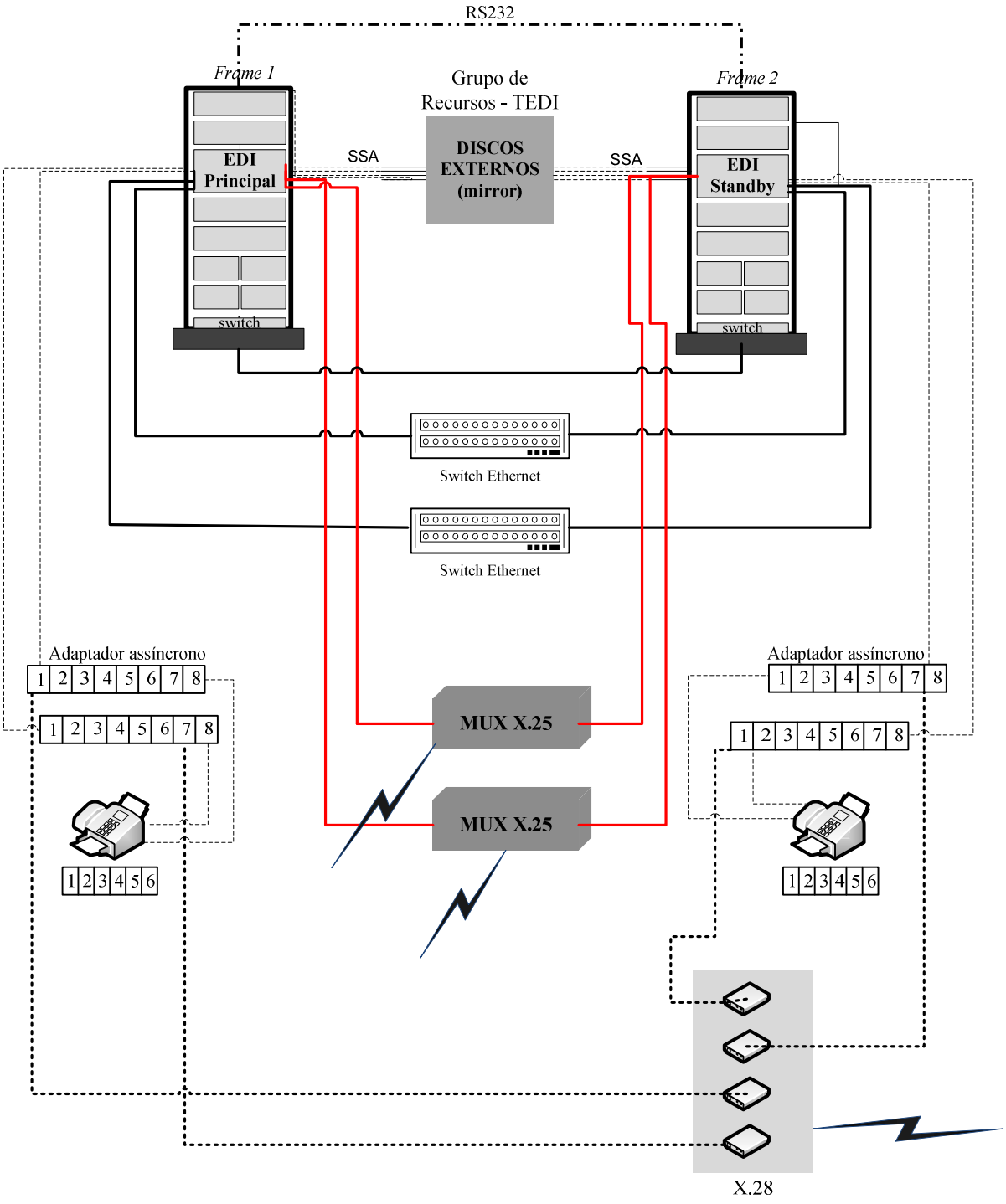


Figura 4-3 – Arquitectura global de alta disponibilidade no projecto TEDI.

Além do servidor como ponto crítico de falha, solucionado com a implementação de um cluster de alta disponibilidade, este projecto apresenta outros componentes igualmente importantes e que são apresentados de seguida. Alguns pontos de falha são comuns a vários projectos, como por exemplo as interfaces de rede. No entanto, há outros que são mais específicos deste projecto, como o equipamento de envio de faxes. No Anexo H pode consultar-se uma proposta de documentação a seguir durante o planeamento da solução final.

4.3.1 A REDE LOCAL

Para eliminar este ponto crítico de falha foram tomadas as seguintes acções:

- As duas placas de rede Ethernet, uma no nó principal e outra no de *standby*, asseguram o acesso aos nós, a partir do exterior do SP.
- A Ethernet interna e o *switch* do *SP* existem apenas em configurações do tipo SP. Permitem que os nós, por pertencerem a um outro cluster, o SP, estejam submetidos a um tratamento privado em termos de gestão.
- A ligação RS232 entre os dois nós do cluster pretende reduzir significativamente os falsos positivos que possam ocorrer, evitando transferências desnecessárias dos grupos de recursos para o nó de *backup.* Se ocorrer um problema com a rede Ethernet que não seja ao nível físico, a retoma de serviços não resolverá o problema. Assim, a ligação série assegura a continuidade dos serviços mesmo no caso de falha da rede Ethernet. Não é obrigatório que a ligação entre os nós seja do tipo RS232, podendo igualmente usar-se o protocolo SCSI ou outro, sendo que para o cluster se trata de mais uma rede definida que não tem qualquer tratamento especial em relação às outras. Esta ligação é efectuada a dois adaptadores assíncronos, ligados um a cada sistema, e acedidos pelo HACMP através de um dispositivo lógico `tty`. Não se considerou haver necessidade em duplicar esta ligação uma vez que continua a haver ligação entre os dois nós do cluster em caso de avaria deste componente, por exemplo através da Ethernet. Se ocorrer simultaneamente uma falha em todas as interfaces de rede (incluindo a do tipo série), então o problema é bem mais grave e complicado e, provavelmente, estará relacionado com outros componentes, nomeadamente com o servidor.

Em termos de acesso externo, existem duas interfaces Ethernet em cada nó, o que significa que no caso de avariar uma delas ou um cabo de rede, a outra placa assume os serviços, não havendo paragem para os utilizadores. Se avariarem ambas as placas externas, os nós do cluster continuam a comunicar através do *switch* e da ligação série, embora não haja acesso do exterior.

Foram usadas duas ligações a segmentos de rede Ethernet independentes, suportados por dois *switches* distintos, conforme se ilustra na Figura 4-3. Esta solução elimina dois pontos críticos de falha; a rede física e os equipamentos activos (*switches* de Ethernet).

4.3.2 O PROTOCOLO IP

A solução apresentada assenta numa rede TCP/IP. Devido ao facto de haver várias interfaces de rede associadas a cada nó do cluster e podendo cada uma ter vários endereços IP, é necessário planear cuidadosamente a atribuição dos endereços a cada interface de rede. A forma mais simples é definir várias sub-redes, cada uma para os endereços das interfaces de serviço e de *standby* de cada nó do cluster. O planeamento da rede IP seguiu algumas regras fundamentais:

- Utilizar redes privadas IP.
- Definir o número de endereços necessários para todas as interfaces de rede do cluster.

- Dimensionar as sub-redes com tamanho suficiente que permita a escalabilidade do cluster. O cenário em causa utilizou apenas dois nós (um principal e outro de *standby*), mas é possível definir clusters mais complexos e com maior número de servidores.
- Definir os endereços reservados em cada sub-rede.
- Calcular a máscara de sub-rede apropriada.
- Acautelar que as interfaces de *standby* e de serviço estão em sub-redes distintas. As interfaces de *standby* deverão estar na mesma sub-rede IP.
- Configurar o endereço de *boot* na placa de serviço e assegurar que existe na mesma sub-rede que o endereço de serviço do nó primário.

4.3.3 OS ADAPTADORES ASSÍNCRONOS

Um adaptador assíncrono é um dispositivo de *hardware* que contém várias ligações RS232. Cada adaptador encontra-se ligado a uma porta série dos nós do cluster. Neste cenário, para suportar a ligação de periféricos externos, como o equipamento de fax, os modems e a ligação RS232, foram utilizados adaptadores assíncronos de 8 portas. A cada porta deste adaptador foi atribuído um endereço (`tty`) com as características desejáveis de acordo com o periférico em causa. Como se trata de equipamentos que podem avariar, comprometendo o normal funcionamento do cluster (e consequentemente, do negócio), foram instalados em cada nó do cluster dois adaptadores assíncronos. Também os periféricos que estão ligados a estes adaptadores foram instalados em duplicado. Assim, um determinado periférico está ligado a um adaptador de cada nó para que se assegure a falha do nó, do adaptador e também do periférico.

Para simplificar a gestão e operação do cluster, os dispositivos `tty` criados no sistema operativo ficaram com o mesmo nome nos dois nós do cluster, quando associados ao mesmo tipo de periférico. A ligação série entre os nós foi igualmente acoplada a estes dispositivos assíncronos e a sua identificação foi a mesma em ambos os nós.

4.3.4 OS DISCOS

Cada nó tem dois discos internos, um como *mirror* do outro. Além do sistema operativo incluíram-se nestes discos as aplicações não críticas que não faziam parte do grupo de recursos definido no cluster. No nó principal foi apenas incluída a partição lógica com o sistema operativo AIX (`rootvg`). Por outro lado, no nó de *standby*, foram igualmente instalados os componentes aplicacionais de desenvolvimento, de que não haverá retoma em caso de falha. Esta opção permitiu utilizar o nó de *standby* exclusivamente para o desenvolvimento e testes, sempre que não estivesse a suportar os serviços do cluster em caso de falha do nó principal. No entanto, foi assumido que, quando este nó realizasse a retoma das aplicações do cluster, os utilizadores que se encontrassem a trabalhar no ambiente de desenvolvimento seriam automaticamente desligados, passando este a dedicar-se por inteiro às aplicações do cluster. O processo de retoma de serviços tinha necessariamente maior prioridade que os utilizadores que, eventualmente tivessem sessões interactivas no ambiente de desenvolvimento e testes.

De uma forma geral, em aplicações de alta disponibilidade, nos discos internos de cada nó é sempre aconselhável incluir apenas o *software* e dados associado a aplicações não críticas (por exemplo, bases de dados de desenvolvimento e testes, versões de teste do *software* que se encontra em desenvolvimento) e, por outro lado, dos discos externos devem apenas constar as aplicações e os dados associados aos grupos de recursos dos quais se pretenda a retoma em caso de falha.

Os discos externos alocados ao cluster foram ligados aos nós por SSA e configurados fisicamente em *mirroring*. A capacidade alocada ao projecto foi dimensionada para suportar as várias aplicações em causa, nomeadamente o tradutor de EDI, a aplicação de controlo de faxes, a de transferência de ficheiros para a VAN, o sistema gestor de base de dados (Oracle) e os dados propriamente ditos.

A identificação no AIX dos discos internos e externos (`pdisk` e `hdisk`[5]) foi definida igualmente em ambos os nós, definindo-se também uma nomenclatura para a adição de novos discos no sistema. Após a substituição de um disco avariado, o sistema reconfigura normalmente o novo disco com novos identificadores lógicos, tendo-se definido um procedimento (sob a forma de *script*) que colocava as definições iguais em ambos os nós e de acordo com a nomenclatura adoptada. No Projecto TEDI os discos externos foram partilhados por ambos os nós de uma forma não concorrente. Ou seja, o nó de *standby* apenas acedia aos dados no caso de retoma dos serviços do cluster em caso de falha, quando o nó primário já tivesse libertado esses mesmos recursos.

4.3.5 O ACESSO À REDE PÚBLICA DE DADOS

As comunicações com o exterior, nomeadamente o envio de dados para a VAN, utilizaram uma infra-estrutura de acesso à rede pública X.25. Em cada nó foram instalados dois acessos X.25, ligados a comutadores X.25 distintos. Eliminaram-se assim dois pontos críticos de falha importantes: o adaptador instalado no nó e o comutador de X.25. Para ambos os casos foi possível ter uma ligação alternativa em caso de falha.

A dependência à rede pública de dados foi considerada de criticidade elevada neste projecto. Assim, foi criada uma segunda linha de redundância à ligação X.25 existente, tendo sido instalados quatro modems para acesso ao exterior, de uma forma assíncrona via X.28. Estes modems foram ligados também aos concentradores assíncronos e configurados através de um dispositivo `tty`**.** A infra-estrutura de comunicação com o exterior foi assim dotada de caminhos alternativos, activados automaticamente em caso de falha da ligação principal X.25. A aplicação de envio automático dos dados para a VAN foi adaptada para tirar partido desta vantagem, escolhendo de forma automática a ligação ao exterior que se encontrava activa em cada momento.

4.3.6 O EQUIPAMENTO DE FAX

Alguns fornecedores não tinham implementado um tradutor de EDI para o seu sistema de informação, não tendo por isso acesso à VAN. Por outro lado, havia igualmente fornecedores que não tinham nenhum sistema de informação implementado, podendo apenas receber as encomendas

[5] `pdisk` representa a identificação dos discos físicos. `hdisk` identifica logicamente os discos físicos (`pdisk`). O sub-sistema de gestão de partições do AIX (LVM) pode assim associar os pdisk ou hdisk a *filesystems* ou directamente às aplicações (por exemplo pelo SGBD).

em papel ou por fax. Nesse sentido, um dos requisitos consistia na necessidade de implementar uma infra-estrutura aplicacional que permitisse, não só o envio automático dos dados através de um fax, mas também o registo sobre o sucesso ou insucesso do envio das mensagens.

Por se tratar de um ponto crítico de falha, foram utilizados dois equipamentos de fax, cada um dispondo de 6 linhas de envio. Estes equipamentos foram ligados aos concentradores assíncronos e identificados com um dispositivo `tty`. Cada equipamento estava dotado de duas placas assíncronas que permitiram ligar duas linhas de fax, uma para envio e outra para monitorização do estado. Assim, no caso de avaria numa placa do equipamento de fax, este serviço seria imediatamente retomado pela placa gémea do equipamento de fax, no mesmo adaptador assíncrono. No caso de uma avaria no adaptador assíncrono, os serviços seriam assegurados pelo equipamento de fax que se encontrava ligado ao adaptador assíncrono de *backup* no mesmo nó. A duplicação dos adaptadores assíncronos e dos equipamentos de fax no nó de *standby* permitiu assegurar igualmente a continuidade do envio de faxes no caso de retoma dos grupos de recurso para o nó de *standby*. Assim, foi possível ter vários caminhos alternativos para evitar falhas ao nível do envio de faxes para o exterior. Também neste caso foi necessário adaptar os programas de envio de faxes para monitorizar qual a ligação activa em cada momento e tirar assim partido das comutações automáticas verificadas em caso de falha de *hardware*.

4.3.7 A FRAME DO SP

Os nós alocados ao projecto TEDI estavam acoplados na infra-estrutura do SP, conforme se ilustra na Figura 4-4. Relativamente à ligação eléctrica, ventilação e acesso ao *switch* de elevada performance, as *frames* eram independentes entre si. Ou seja, a paragem de uma *frame* (e dos nós que sustentava) não implicava a paragem dos nós da outra *frame* [15, 16, 62]. Os nós foram instalados fisicamente em duas *frames* distintas, eliminando assim a *frame* como ponto crítico de falha.

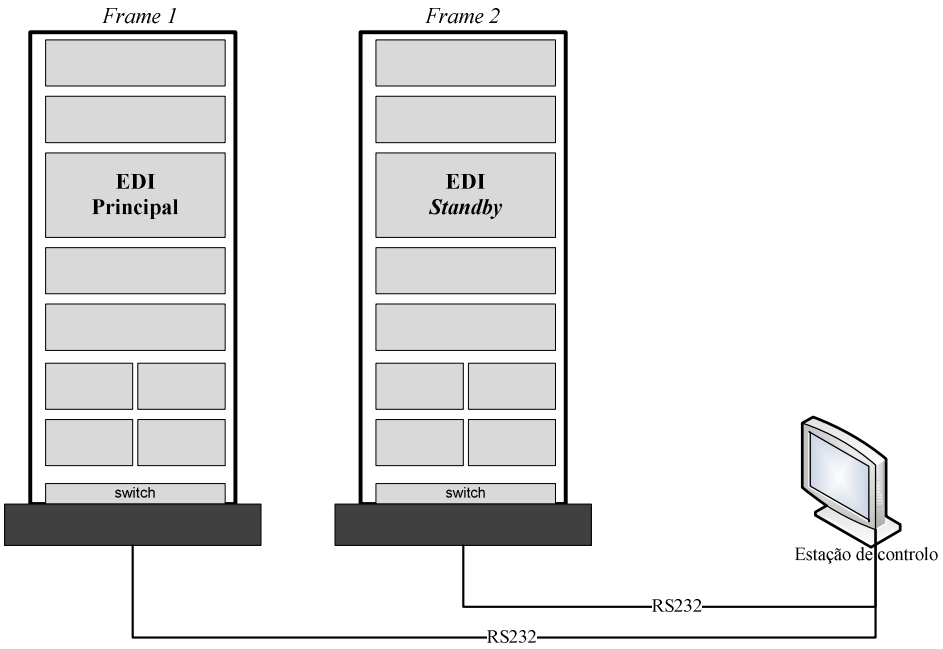


Figura 4-4 – Integração dos nós do TEDI na arquitectura IBM SP.

Esta estratégia permitiu, por um lado aproveitar as funcionalidades de alta disponibilidade associadas à tecnologia SP e por outro eliminar a *frame* como ponto crítico de falha. Em termos de localização, ambas as *frames* residiam na mesma sala. Inicialmente sem nenhuma divisão entre elas, o que não era a solução ideal. Mais tarde, as *frames* foram isoladas fisicamente, aumentando assim o nível de disponibilidade do cluster e a possibilidade de recuperação após a destruição das instalações.

4.3.8 OS GRUPOS DE RECURSOS

A estratégia adoptada para a retoma dos serviços pelos nós do cluster foi em cascata, com acesso não concorrente e sem *mutual takeover*. No caso concreto do projecto TEDI, seria indiferente utilizar uma estratégia em cascata ou em rotação. No entanto, atendendo à escalabilidade prevista para o cluster, fazia sentido que, em caso de falha e após a operacionalidade do nó primário, fosse este a recuperar os serviços, sob pena de perda de desempenho no cluster.

Relativamente ao acesso aos discos, foi definido o acesso não concorrente, uma vez que não haveria *mutual takeover* de aplicações. O uso desta estratégia de retoma dos recursos é aconselhável quando os dois nós do cluster são utilizados para realizar processamento efectivo e, em caso de falha, um dos nós retoma todos os serviços do cluster. Neste caso, perde-se o conceito original de nó principal e de *standby*. Um exemplo típico deste tipo de configuração é a utilização da versão paralela do Oracle (*Oracle Parallel Server*), com uma instância da base de dados em execução em cada nó com acesso concorrente aos discos externos, podendo ambos os nós recuperar os recursos em caso de falha.

Um grupo de recursos, como já foi explicado anteriormente, deve ser entendido como o conjunto de componentes lógicos do cluster que acompanha a paragem e o arranque de um nó em caso de uma falha que origine uma retoma de serviços. Quando ocorre uma falha no nó principal que origine essa retoma, é realizada a transferência dos endereços IP e MAC da interface de serviço (Secção 4.2), e de seguida são transferidos todos os componentes do grupo de recursos. Assim, deverão ser incluídos no grupo de recursos todos os componentes que acompanham essa transferência. No projecto TEDI foi definido apenas um grupo de recursos constituído por:

- Partição lógica dos discos externos onde residem as aplicações do negócio.
- *Filesystems* configurados nos discos externos.
- Interface de rede de "serviço" do cluster.
- *Scripts* de paragem e arranque das aplicações de negócio envolvidas.

Em [30, 55] é possível encontrar o detalhe dos procedimentos envolvidos na configuração de um cluster de HACMP, nomeadamente os tipos de retoma dos grupos de recurso e as variantes possíveis. Em [48] estão documentadas todas as operações de configuração do cluster HACMP levadas a cabo para o projecto TEDI.

4.4 NOTIFICAÇÃO AUTOMÁTICA

Esta não é propriamente uma funcionalidade do HACMP. Com efeito, o AIX permite a execução de um *script* quando ocorre um determinado evento no sistema. Esta funcionalidade não deve ser confundida com a possibilidade que o HACMP tem de criar novos eventos e associar-lhes um *script*, conforme descrito na Secção 4.2.

O HACMP, como qualquer outra aplicação de alta disponibilidade em clusters, não gere falhas aplicacionais. É necessário um trabalho adicional de monitorização das aplicações e notificação automática, a fim de minimizar o tempo de resposta a eventos aplicacionais críticos. No projecto TEDI este trabalho realizou-se ao nível do sistema gestor de base de dados Oracle e da aplicação do tradutor de EDI, tendo-se desenvolvido vários *scripts* de monitorização que implementavam mecanismos automáticos de notificação da equipa de gestão da rede.

Em geral, os sistemas Unix registam as mensagens de erro e aviso relativas a problemas de *hardware* ou falhas aplicacionais, num ficheiro de *log*, que no caso do AIX é o ficheiro `/var/adm/errlog`. As mensagens são identificadas pelo tipo e classe a que pertencem e têm associada uma descrição, ou código, de causas prováveis, bem como as acções a tomar para solucionar o problema. Este ficheiro tem um formato próprio e pode ser lido, de uma maneira formatada e amigável, através do comando `errpt`.

No projecto TEDI foi definido um *script*, escalonado na lista de `cron`[6] do sistema operativo, que verificava se as aplicações em causa estavam a funcionar correctamente e, se não fosse o caso, enviava uma mensagem, através do comando `errlogger`, para o ficheiro `/var/adm/errlog`. No Anexo D é apresentado o *script* relativo ao teste de operacionalidade da base de dados Oracle. A Figura 4-5 apresenta um exemplo da associação de um *script* a um evento do cluster, utilizando a interface de configuração do HACMP.

```
                          Change/Show a Notify Method
Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                                      [Entry Fields]
* Notification Object Name                          EDI_ALERT_HW
* Persistence across system restart?              Yes                    +
  Process ID for use by Notify Method             [0]                    +#
  Select Error Class                              Hardware               +
  Select Error Type                               PERM                   +
  Match ALERTable errors?                         None                   +
  Select Error Label                              []                     +
  Resource Name                                   []                     +
  Resource Class                                  []                     +
  Resource Type                                   []
*Notify Method                          [Alert.sh $1 $2 $3 $4 $5 $6 $7 $8]
```

Figura 4-5 – Exemplo da atribuição de um script a um evento do HACMP.

[6] `cron` corresponde a uma tabela com o escalonamento temporal de tarefas em sistemas Unix.

Neste caso foi definida uma notificação, designada "EDI_ALERT_HW" que filtrava todos os erros de *hardware* e do tipo permanente que ocorressem no sistema. O *script* "Alert.sh" recebia os parâmetros do `errlog` do sistema e validava o tipo de falha que ocorreu e qual o impacto na disponibilidade do cluster. Se a falha estivesse relacionada com o tradutor de EDI então seria desencadeado automaticamente um conjunto de operações de recuperação e notificação das equipas técnicas. Os parâmetros utilizados na execução do *script* "Alert.sh" referem-se aos utilizados pelo comando `errlogger` que escreve directamente para o `errlog` do sistema.

Embora a monitorização das aplicações e a sua recuperação em caso de falha não seja uma tarefa dos programas que controlam o cluster, as funcionalidades de notificação de eventos, associada à implementação de *scripts* específicos para reacção em caso de falha podem constituir um contributo para a extensão das acções do HACMP à componente aplicacional. No caso do TEDI, a estratégia adoptada neste domínio surtiu o efeito desejado.

4.5 TAREFAS COMPLEMENTARES DE GESTÃO

Além da configuração do HACMP, há ainda um conjunto de tarefas de administração de AIX que estão directamente ligadas à instalação e configuração do cluster de alta disponibilidade. Descrevem-se de seguida as que foram levadas a cabo neste projecto. De acordo com a especificidade do projecto, poderão ser identificadas outras tarefas de pós-instalação e de configuração, relacionadas com o contexto em causa.

1. **Aplicações**
   - Definir o nome e localização dos ficheiros de *log*.
   - Construir *scripts* de paragem e arranque das aplicações (Anexo E).
   - Definir políticas de distribuição das últimas versões dos *scripts* e da configuração do HACMP. No caso concreto do SP tirou-se partido do componente *File Collection* do PSSP [18], que permite a propagação distribuída dos ficheiros pelos vários nós do SP.
   - Instalar aplicações: base de dados Oracle, aplicação do tradutor de EDI, ambiente de desenvolvimento no nó de *standby*, aplicações diversas de monitorização.
2. **Gestão de disco**
   - Dimensionar e alocar o espaço de disco para cada aplicação, usando estratégias de *stripping*.
   - Definir a nomenclatura associada aos discos, partições e *filesystems.*
   - Criar partições lógicas e disponibilizá-las a ambos os nós do cluster.
   - Configurar *mirroring* nos discos internos e externos e sincronizar as cópias.
   - Inibir a activação automática das partições lógicas, passando o controlo para o HACMP.
3. **Configurações de rede**
   - Definir os dispositivos lógicos (`tty`) associados às ligações série.

- Configurar as ligações X.25 (canais lógicos e números de identificação, entre outros parâmetros).
- Definir e configurar os endereços IP para as interfaces associadas ao HACMP.
- Incluir os endereços IP no serviço de resolução de nomes (DNS).

4. **Parâmetros gerais do AIX**
   - `maxpout` e `minpout`: número máximo e mínimo de escritas pendentes para disco. Baseado na experiência considerou-se como valores razoáveis o de 32 para o parâmetro `maxpout` e 24 para o `minpout`.
   - `maxuproc`: número máximo de processos que podem ser executados simultaneamente por cada utilizador. Para contemplar um requisito do software aplicacional, este parâmetro foi aumentado para 10000. As alterações podem efectuar-se com um único comando do AIX:

     ```
     chdev –l sys0 –a maxuproc='10000' –a maxpout='32' –a minpout='24'
     ```

5. **Criar utilizadores**
   - Centralmente no SP e distribuídos posteriormente por todos os nós através do PSSP.
6. **Alterar as listas de processos escalonados (`cron`)**
   - Assegurar que as listas de processos escalonados pelos utilizadores através da `cron` são igualmente retomadas pelo HACMP. Para o efeito foi criado um ficheiro com o formato interpretado pelo comando `crontab` numa directoria pertencente ao grupo de recursos. Em caso de retoma, o *script* de arranque das aplicações interpreta esse ficheiro através do comando `crontab < cronfile`.
   - Na paragem do HACMP as tabelas de `cron` dos utilizadores são inicializadas através do seguinte comando:

     ```
     su – <login> –c "crontab < cronfile_vazio"
     ```

   - No arranque foi inicializado o ficheiro de `cron` de cada utilizador:

     ```
     su – <login> –c "crontab < cronfile_user"
     ```

7. **Assegurar a comunicação entre os nós**
   - Incluir no ficheiro `.rhosts` os nomes dos nós do cluster e o acesso autorizado para `root`. Desta forma facilita-se a comunicação entre os nós do SP, incluindo as disponibilizadas no PSSP.

4.6 VALIDAÇÃO DA SOLUÇÃO

Após a configuração do cluster foram realizados vários testes experimentais de aceitação da solução, antes da passagem a produção. Nesta fase foram testados os principais cenários que poderiam ocorrer e documentados os resultados obtidos [48]. Além do teste dos cenários, nesta fase pretendeu-se ainda validar a solução global, identificar alguns ajustes que pudessem realizar-

se à configuração do cluster e ainda afinar os procedimentos de contingência associados a cada evento.

4.6.1 FALHA DO SISTEMA PRINCIPAL

Neste cenário provocou-se uma paragem anormal do sistema primário, tendo-se observado a seguinte sequência de eventos:

- O nó de *standby* deixou de receber mensagens do nó principal.
- Ao fim de alguns segundos (definido na configuração do HACMP), teve início o processo de retoma dos serviços do nó principal para o de *standby*.
- O nó de *standby* foi então promovido a nó principal e assumiu o controlo do cluster.

Para o utilizador final houve apenas uma quebra de serviço correspondente ao tempo necessário para o nó de *standby* adquirir os recursos do cluster. O Anexo G apresenta os *logs* de actividade do HACMP durante o período de transição dos recursos do nó primário para o nó de *standby*.

4.6.2 FALHAS DIVERSAS NA REDE

Para testar a robustez do cluster foram injectadas várias falhas nos componentes de rede, correspondentes às situações mais comuns:

- Desligar o cabo de ligação à interface de rede de serviço do nó principal.
- Desligar o cabo de ligação à interface de rede de *standby* do nó principal.
- Desligar o cabo de ligação à interface de rede de serviço do nó de *standby.*
- Desligar o cabo de ligação à interface de rede de *standby* do de *standby.*
- Desligar a rede externa, a interna, o *switch* e a ligação RS232 no nó principal.
- Desligar o acesso principal à rede pública de dados X.25.
- Desligar a rede constituída pelas interfaces RS232 que suportam a ligação entre os dois nós.

A. INTERFACE DE SERVIÇO DO NÓ PRINCIPAL DESACTIVADA

Neste cenário foi gerado um evento de paragem da interface de serviço no nó principal. A interface de rede de *standby* do nó principal foi configurada com o endereço IP de serviço. Do ponto de vista dos utilizadores não houve lugar a qualquer quebra de serviço, passando-se a aceder no entanto ao nó primário por uma nova interface de rede.

B. INTERFACE DE SERVIÇO DO NÓ DE *STANDBY* DESACTIVADA

Neste segundo cenário foi gerado o evento “service_adapter_down” no nó de *standby* do cluster. A interface de *standby* do nó de *standby* foi configurada com o endereço de serviço,

assegurando que uma eventual falha do nó principal continua a ser assegurada pelo nó de *standby*. Este evento não afectou o normal funcionamento do cluster e a sua disponibilidade para os utilizadores.

C. AMBAS AS INTERFACES DE REDE EXTERNAS DESACTIVADAS NO NÓ PRINCIPAL

Neste caso ocorreu um evento de "network_down" no cluster. Não houve lugar a qualquer retoma de serviços, uma vez que continuou a existir conectividade entre os nós a partir da rede interna e também pela ligação RS232. No entanto, deixou de ser possível aceder ao nó principal (e ao cluster) a partir do exterior. A rede interna que interliga os nós no SP manteve a sua conectividade. Os processos de monitorização detectaram este evento e para os utilizadores houve lugar a uma quebra de serviço.

D. REDES EXTERNA, INTERNA E SÉRIE DESACTIVADAS NO NÓ PRIMÁRIO

A avaria ou desactivação forçada e simultânea de todas as interfaces de rede são cenários invulgares e pouco frequentes. Corresponde ao cenário que não se pretende que aconteça, já que o nó de *backup* tenta realizar a retoma dos recursos do cluster, sem que estes tenham sido libertados pelo nó principal. Esta situação é a mais incómoda numa configuração de alta disponibilidade num cluster já que, por um lado é uma situação que provoca falha dos serviços ao utilizador e por outro, a retoma automática dos serviços pelos nós do cluster não irá resolver o problema. Para mitigar o impacto deste tipo de problemas, a solução passa por definir estratégias de monitorização mais exigentes que tentem identificar a ocorrência de falsos positivos que provoquem a retoma desnecessária dos serviços do cluster.

E. LIGAÇÃO SÉRIE DESACTIVADA

Os nós deixaram de ser visíveis através da ligação série, mas a comunicação a partir das restantes interfaces de rede continuou activa. Para os utilizadores não se registou nenhuma quebra de serviço, embora o cluster tenha ficado sem protecção contra uma falha da rede Ethernet.

F. PROBLEMAS COM ADAPTADOR DE PORTAS ASSÍNCRONO

Este dispositivo alojava vários equipamentos, como os modems X.28 e o equipamento de fax. Foi desenvolvida uma pequena aplicação de monitorização do estado de cada uma das portas e interfaces, no sentido de detectar alguma anomalia no adaptador assíncrono e, nesse caso, proceder à utilização de outro alternativo. Para o cluster não houve registo de qualquer evento, nem tão pouco para os utilizadores. Os serviços suportados pelos equipamentos ligados ao adaptador de portas assíncrono, nomeadamente o envio de faxes, também não sofreram qualquer quebra, porque a detecção da anomalia encaminhou os faxes para o adaptador alternativo.

G. AVARIA DE INTERFACES DE REDE EXTERNAS E/OU INTERNAS

Quando há uma interface de rede que não está operacional é gerado um evento do cluster que efectua a transferência dos endereços IP e de MAC para uma outra interface, não havendo a retoma dos serviços para outro nó. Se ambas as placas de uma determinada rede definida no cluster ficam indisponíveis, há lugar a um evento de "network_down" e neste caso os utilizadores poderão não conseguir aceder ao servidor. Como no cluster do projecto TEDI existe uma ligação série activa, os nós continuam a comunicar, não havendo por isso lugar a nenhuma retoma dos recursos. Caso contrário, esta situação daria lugar a um falso positivo, dando assim origem a uma falsa retoma dos recursos do cluster.

4.6.3 SUBSTITUIÇÃO DE *HARDWARE* AVARIADO

A substituição de *hardware* é uma das operações onde o uso de um cluster de alta disponibilidade pode minimizar o período de paragem dos serviços para o utilizador. Esta operação pode, ou não, causar quebra do serviço aos utilizadores, dependendo do componente e do nó em causa. No caso de um componente do nó principal, então a solução consiste em:

- Transferir os recursos para o nó de *standby*.
- Desligar os serviços de HACMP.
- Desligar electricamente o nó primário e substituir o componente.
- Ligar o nó principal. No arranque do sistema são iniciados os serviços de HACMP, o que origina uma retoma dos recursos novamente para o nó primário, convergindo o cluster para a sua forma inicial.

Para os utilizadores esta interrupção resumiu-se apenas ao período de tempo necessário para a transferência dos recursos entre os nós. No entanto, estas paragens devem ser, sempre que possível, planeadas e os utilizadores avisados. No caso de o componente residir no nó de *standby*, os serviços de HACMP têm de ser desligados neste nó, mas não há lugar a retoma dos recursos e os utilizadores continuarão a aceder às aplicações.

4.6.4 AVARIA NOS DISCOS PARTILHADOS

As unidades de disco externas foram configuradas com *mirror*. No caso de ocorrer uma avaria num disco externo, a sua cópia assegura o acesso aos dados, não havendo assim paragem dos nós e quebra de serviço para os utilizadores. No entanto, a substituição do disco avariado e a correspondente sincronização da cópia de *mirror* deve ser realizada o mais rapidamente possível.

4.6.5 AVARIA NO EQUIPAMENTO DE FAX

Foram injectadas várias falhas no equipamento de gestão de fax, verificando-se em todos os cenários uma comutação automática para o equipamento de *backup.*

4.6.6 OPERAÇÕES DE BACKUP

As operações de *backup* não tiveram impacto significativo na actividade do cluster, não se tendo registado nenhuma quebra de serviço. Relativamente ao *backup* imagem do sistema operativo, foi realizado após a estabilização do cluster, com a seguinte sequência de operações:

- Realizar *backup* imagem ao nó de *standby*
- Efectuar a paragem dos serviços do HACMP com a opção "graceful with takeover", havendo transferência dos recursos do nó principal para o de *standby*.
- Realizar o *backup* imagem do nó primário, agora em modo de *standby*.

Quanto ao *backup* das aplicações, identificaram-se três tipos de dados a salvaguardar:

- Programas, especialmente o tradutor de EDI e os scripts de administração do cluster.
- Ficheiros de *log* e da aplicação HACMP.
- Dados do negócio armazenados na base de dados Oracle.

Os dois primeiros tinham uma periodicidade diária. Relativamente aos dados da base de dados, optou-se pelo *backup offline* inicial e pelo *export on-line* diário.

4.6.7 PARAGEM ANORNAL DAS APLICAÇÕES DE NEGÓCIO

As paragens das aplicações específicas do projecto TEDI não foram controladas directamente pelo HACMP. Para tal, foram desenvolvidos *scripts* de monitorização desenvolvidos para o efeito (Anexo B, Anexo C e Anexo D) que geram eventos no *log* do sistema operativo e enviam uma notificação automática à equipa de administração do cluster. Verificou-se a obrigatoriedade de existirem *scripts* que automatizassem os processos de paragem, arranque e monitorização das aplicações. Relativamente ao HACMP, o impacto nos utilizadores dependeu sempre da forma como os processos foram terminados, conforme descrito na Secção 4.2.

4.7 ESCALABILIDADE DO CLUSTER

A solução apresentada para o projecto TEDI previa a inclusão a curto prazo de novas aplicações no cluster de HACMP. Tratava-se do projecto de Gestão de Inventário (GI), que incluía aplicações críticas e que seriam naturalmente eleitas para serem integradas no cluster.

Genericamente, este projecto partilhou o nó de *backup* utilizado pelo projecto TEDI e usou um novo nó primário, em fase de aquisição. Assim, no limite, o actual nó de *backup* poderia necessitar de retomar simultaneamente as aplicações do TEDI e do GI. A arquitectura global de alta disponibilidade de ambos os projectos encontra-se ilustrada no Anexo A.

Aproveitando a infra-estrutura existente para o projecto TEDI, encetou-se a implementação do projecto GI, com os seguintes passos principais:

1) Planear a configuração e implementação desta nova aplicação, usando a metodologia abordada neste capítulo. Essencialmente, foi necessário identificar e resolver os novos pontos críticos de falha e as especificidades das aplicações envolvidas.

2) Incluir o novo nó no cluster para o novo projecto. Configurar o nó seguindo a metodologia abordada anteriormente para os nós do projecto TEDI.

3) Efectuar algumas alterações ao cluster, nomeadamente:

   - Definir um novo grupo de recursos dedicado às novas aplicações.
   - Instalar no nó de *standby* do cluster (comum aos dois projectos), mais uma interface de rede Ethernet para contemplar retoma das aplicações de GI. Esta interface foi incluída numa sub-rede IP diferente da do TEDI e correspondeu, em termos de HACMP, a uma interface de *standby* para o cluster GI.
   - Instalar mais uma ligação RS232 que interligou o nó de *standby* e o principal do projecto GI.
   - *Upgrade* à configuração do nó de *standby*. Uma vez que este nó iria retomar as aplicações de dois nós diferentes, teve de se atender à possibilidade de essa retoma ocorrer em simultâneo, ou seja, haver uma falha nos nós primários dos dois projectos ao mesmo tempo.
   - A estratégia relativamente à operação de *takeover* e acesso aos discos externos manteve-se em cascata e não concorrente, respectivamente.

Uma componente importante destes dois projectos foi igualmente a gestão dos *backups* realizados aos dados. Para tal, estes dois projectos foram incluídos na gestão centralizada de *backups* adoptada para todos os servidores geridos pela SRD, descrita no Capítulo 7.

5 CASO DE ESTUDO 2 - ALTA DISPONIBILIDADE EM LINUX

A implementação de clusters de alta disponibilidade em Linux partilha a generalidade dos conceitos gerais descritos na Secção 3.6. Ao contrário do HACMP para AIX (Capítulo 4), as soluções existentes para Linux são *open source* e usam uma licença GPL. As aplicações de alta disponibilidade existentes para Linux aparecem diluídas em vários projectos, que tiveram mais ou menos sucesso nas suas implementações práticas. Um caso de sucesso é o projecto Linux Virtual Server (LVS) [63-65], que se desdobra em inúmeras aplicações para balanceamento de carga usando clusters e também para alta disponibilidade, como é o caso da aplicação ”heartbeat” [66].

A motivação para realizar os testes com esta aplicação assentou em três objectivos principais. Em primeiro lugar, pretendeu-se demonstrar a possibilidade de implementar um cluster de alta disponibilidade recorrendo apenas a aplicações *open source*, como o “heartbeat”, tirando assim partido das características que lhe estão associadas, como os baixos custos de licenciamento, instalação e actualização. Para organizações de pequena dimensão que pretendem implementar soluções de alta disponibilidade a custos controlados, esta pode ser uma solução a explorar. De seguida, pretendeu-se testar a exequibilidade da integração de serviços de rede comuns em clusters de alta disponibilidade, como o HTTP, o Proxy e o e-mail. Por fim, pretendeu-se averiguar as funcionalidades observadas nas aplicações comerciais, como sejam aplicações de monitorização, retoma de grupos de recursos e retoma de endereços IP e MAC. Além dos testes realizados em contexto empresarial, orientei ainda um trabalho de índole académico, onde foram igualmente exploradas e testadas as capacidades do “heartbeat” [47, 67].

5.1 A APLICAÇÃO *HEARTBEAT* PARA LINUX

À semelhança do HACMP, o “heartbeat” implementa um protocolo aplicacional que verifica o estado dos nós do cluster através da troca de mensagens UDP pelas interfaces de rede configuradas. Ao nível da rede, se as interfaces de serviço falharem, as correspondentes de *standby* são automaticamente configuradas para assegurarem o acesso dos utilizadores. Relativamente aos nós, no caso do nó principal perder a conectividade com o cluster, entra em funcionamento o procedimento de retoma das aplicações para o nó de *standby*, assegurando a partir desse momento o acesso por parte dos utilizadores.

A instalação e configuração do “heartbeat” estão amplamente descritas em [47] e na documentação disponibilizada pelo grupo que coordena o desenvolvimento da aplicação [68]. A Figura 5-1 resume a estrutura de directorias criada durante a instalação, destacando-se os ficheiros de configuração (“input”), os executáveis que processam os eventos e tomam decisões (“Execução”) e, por último, os que registam as actividades do cluster (“Output”) [47].

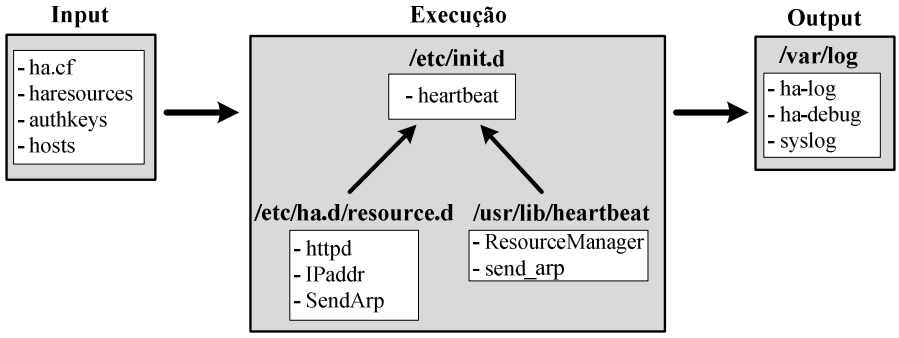


Figura 5-1 – Estrutura de directorias e ficheiros da aplicação "heartbeat".

A inicialização do "heartbeat" lê a informação de configuração dos grupos de recursos do cluster (ficheiros `haresources, authkeys e hosts`) e inicializa o processo responsável pelo processamento dos eventos, o `clusterd`. Na paragem do "heartbeat" (`/etc/init.d/heartbeat stop` ou `service heartbeat stop`) é executada uma função específica que liberta os recursos utilizados e termina todos os processos que lhe estão associados, incluindo o envio de pacotes entre os nós. O processo de paragem do "heartbeat" no nó principal e a consequente libertação dos grupos de recursos, origina uma sequência de arranque das aplicações no nó de *standby*.

5.2 CENÁRIOS DE TESTE

Foram realizadas duas baterias de testes semelhantes, mas em momentos temporais distintos. A primeira decorreu na rede de testes da Novis, com as versões iniciais do "heartbeat". A segunda decorreu mais recentemente com a versão 1.2.3, encontrando-se documentada em [47]. Em termos gerais, os clusters implementados em ambos os testes são muito semelhantes aos descritos no Capítulo 4. São constituídos por dois nós que partilham o mesmo grupo de recursos que é retomado pelo nó de *standby* em caso de falha do nó principal. As aplicações envolvidas no grupo de recursos partilham o mesmo objectivo comum de testar a alta disponibilidade em serviços de rede. Nos testes realizados mais recentemente foram exploradas as aplicações "apache" e "squid", correspondentes aos serviços de rede HTTP e Proxy respectivamente [47, 67].

Tratando-se de cenários de teste, os pontos críticos de falha identificados foram essencialmente os relacionados com a rede. Quanto à topologia do cluster, foram desenvolvidos dois tipos de cenários, conforme ilustrado na Figura 5-2.

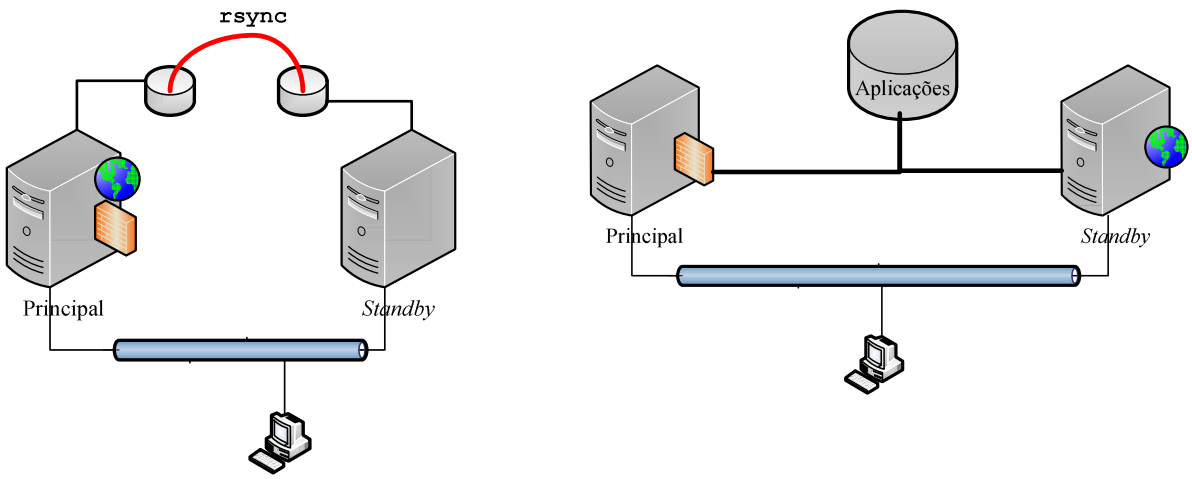


Figura 5-2 – Cenários com clusters de alta disponibilidade usando o "heartbeat".

No primeiro os dados aplicacionais encontravam-se instalados nos discos locais de ambos os nós. Embora não seja um cenário muito flexível, em situações onde a configuração das aplicações geridas pelo cluster seja pouco dinâmica, como por exemplo as páginas HTML disponibilizada pelo servidor HTTP, esta solução poderá ser suficiente e de custo reduzido, bastando apenas implementar mecanismos de sincronização dos dados entre os nós do servidor, como por exemplo através do programa `rsync`. No segundo cenário testou-se igualmente a partilha dos dados aplicacionais através de discos externos, semelhante ao descrito no Capítulo 4.

Foi igualmente testada a possibilidade de utilizar um cluster com três nós, sendo um o nó de *standby* dos dois grupos de recurso, conforme se ilustra na Figura 5-3. Este cenário é semelhante ao descrito na Secção 4.7 que descreve a integração do projecto TEDI com o GI.

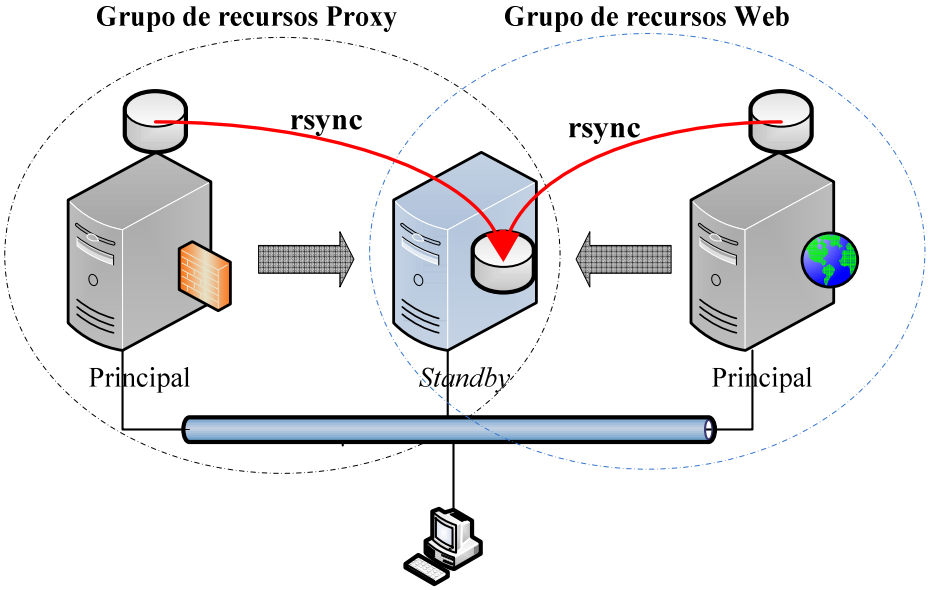


Figura 5-3 – Cenário com um cluster de três nós.

Para a sincronização automática dos relógios dos nós do cluster foi utilizado o protocolo NTP.

5.3 RESULTADOS

Os cenários descritos foram apenas testados em ambiente de teste, não tendo sido integrados em nenhum ambiente de produção. No entanto, os resultados obtidos deram boas indicações sobre a possibilidade de integrar uma solução deste tipo em ambientes de produção. Os testes realizados para validar a solução de *clustering* usando o "heartbeat" englobaram essencialmente a componente de rede e os próprios nós, destacando-se [47]:

- Remover o cabo de rede na interface de serviço do nó principal
- Remover o cabo de rede na interface de serviço do nó de *standby*
- Desligar o nó principal, originando a retoma de recursos para o de *standby*
- Desligar o nó de *standby*

O cenário foi testado num ambiente controlado. A configuração das aplicações foi colocada a um nível mínimo e não houve componentes de *hardware* específicos. No entanto, os testes realizados permitiram comprovar:

- A robustez do "heartbeat" nas operações de retoma dos recursos.
- A possibilidade de desenhar clusters com um nível de complexidade semelhante ao de outras aplicações comerciais, como por exemplo o HACMP, tirando partido das mais-valias associadas ao uso de aplicações *open source*.

Outro aspecto relevante no uso da aplicação "heartbeat" tem a ver com o facto de estar, em certa medida, integrado com outras aplicações do projecto LVS, nomeadamente as relacionadas com o balanceamento de carga, como é o caso do "ultramonkey" (Secção 3.5). Desta forma, torna-se possível implementar um cluster de alta disponibilidade que, simultaneamente, assegure a transferência dos serviços para um servidor de *backup* e efectue o balanceamento dos pedidos realizados aos nós do cluster [31].

6 PROJECTO NPO - ENTREPOSTOS

O projecto NPO-Entrepostos teve início em final de 1998 e foi desenvolvido nos centros de distribuição da empresa Modis, operador logístico do grupo Sonae. Genericamente, o projecto consistiu na implementação de uma nova plataforma operacional dos centros de distribuição, englobando entre outros aspectos uma nova aplicação de gestão e uma renovada infra-estrutura de rede e servidores. Neste capítulo enquadram-se os conceitos principais da gestão de entrepostos e de inventário e descreve-se o projecto, dando-se ênfase às funções que desempenhei e à solução tecnológica adoptada. Em [69] estão documentadas as configurações técnicas realizadas no âmbito do projecto.

A empresa Modis foi criada com o objectivo de garantir a distribuição dos produtos dos centros de distribuição, de agora em diante designados de entrepostos, para a cadeia de lojas existente no grupo Sonae, nomeadamente Modelo, Continente e Worten, entre outras. Dessa forma, assume-se como o operador logístico do grupo Sonae e, além da distribuição da mercadoria às lojas, garante ainda a operação e gestão do espaço nos entrepostos, o planeamento de entregas dos fornecedores, bem como o planeamento e gestão de transportes.

Neste projecto a SRD, juntamente com a Enabler, apareceram como parceiros tecnológicos, garantindo as fases de desenvolvimento, passagem a produção (*roll out*) e estabilização das implementações da solução em cada entreposto. A SRD assumiu a vertente tecnológica do projecto e a Enbaler desempenhou o papel de elemento integrador da solução aplicacional no sistema de informação da Sonae. A Figura 6-1 ilustra a equipa de projecto, destacando as principais entidades envolvidas e as suas responsabilidades.

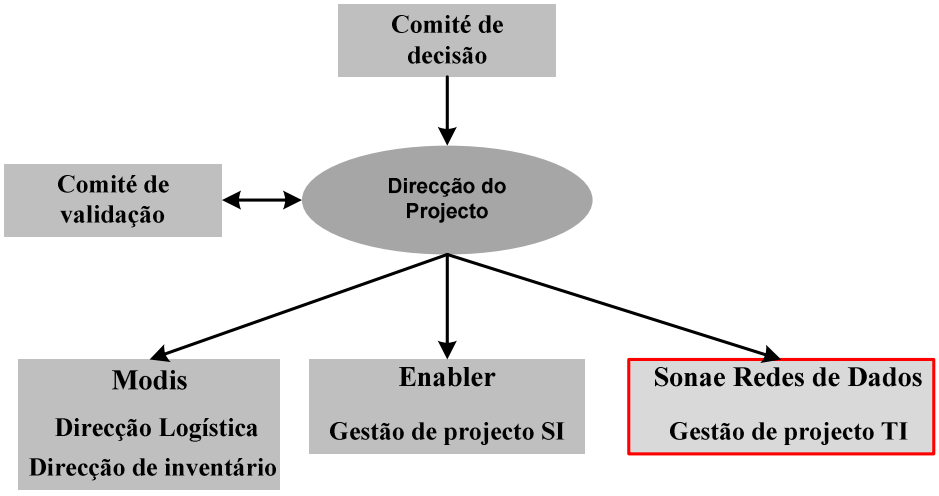


Figura 6-1 – Visão global da equipa do projecto NPO.

Em cada entreposto foram levadas a cabo as seguintes acções principais:

- Implementar a solução aplicacional adquirida.
- Migrar os dados do sistema de informação da gestão de entrepostos, vigente na altura, para o novo.

- Implementar a plataforma tecnológica, onde se destacam os seguintes componentes:
    - Servidores
    - Arquitectura de alta disponibilidade
    - Plano de contingência, segurança e salvaguarda dos dados
    - Rede de rádio frequência no entreposto
- Configurar e testar todos os componentes tecnológicos do projecto.
- Formar os elementos das várias entidades nas respectivas áreas do projecto.
- Desenvolver e implementar aplicações de monitorização dos componentes críticos do projecto.
- Definir e acompanhar o modelo de suporte ao projecto.

As responsabilidades da SRD no projecto, do qual assumi a gestão e coordenação, centraram-se na plataforma tecnológica TI, destacando-se as seguintes:

- Apoiar na escolha da arquitectura tecnológica.
- Instalar o sistema Unix com os requisitos aplicacionais definidos.
- Implementar a arquitectura de alta disponibilidade usando um cluster HACMP.
- Definir o plano de contingência da plataforma tecnológica do projecto, destacando-se a definição de uma bateria de testes periódicos aos componentes, testes de aceitação da solução de alta disponibilidade e da estratégia de *disaster recovery* adoptada.
- Desenvolver mecanismos de monitorização e gestão proactiva do sistema.
- Formar operadores das diversas entidades nos vários componentes do projecto.
- Implementar, configurar e testar os componentes tecnológicos do entreposto, onde se destacam a rede de rádio frequência e todos os terminais que nela se apoiam.
- Implementar o modelo de suporte definido no âmbito do projecto.

A direcção e coordenação geral do projecto foram da responsabilidade da Modis, cabendo-lhe entre outras missões, a especificação de requisitos, a tomada de decisão sobre a aquisição dos componentes tecnológicos, a parametrização da aplicação e a validação final. A equipa de projecto da Modis era multidisciplinar, incluindo elementos de diversas áreas, como a Direcção Logística e a de Inventário.

Quanto à Enabler, assumiu o papel de integrador aplicacional, destacando-se nas suas funções a elaboração dos requisitos funcionais com base nas especificações, a migração do sistema de informação existente para a nova aplicação de gestão de entrepostos e a garantia dos processos de transição.

6.1 ÂMBITO

As acções atrás descritas foram realizadas em todos os entrepostos da Sonae Distribuição. Além dos entrepostos principais foram também considerados os armazéns satélites com utilizações muito específicas. Para suportar a nova plataforma aplicacional foram utilizados três clusters de sistemas Unix (AIX) com uma arquitectura de alta disponibilidade HACMP, instalados nos entrepostos mais importantes: Maia, Azambuja e Alverca. Cada cluster acolheu uma instalação da aplicação de gestão de entreposto, para cada entreposto suportado, conforme a lista descrita na Tabela 6-1.

Cluster	Entrepostos	Tipo
Maia	Maia	Produtos alimentares (com PBL)
	Valadares	Frescos
Azambuja	Azambuja	Produtos alimentares (com PBL) e Frescos
	Instalações da Frisul	Congelados
	Vila Nova da Rainha	Sazonal (campanhas)
Alverca	Alverca 1	Domicilio Bazar Pesado
	Alverca 3	Bazar Têxtil
	Alverca 4	Bazar Ligeiro

Tabela 6-1 – Lista de entrepostos incluídos no projecto NPO-Entrepostos.

Os seja, o sistema da Maia assegurou duas instalações da aplicação e os de Azambuja e Alverca asseguraram três cada um, respectivamente.

A Figura 6-2 ilustra a visão geral da rede, com a distribuição espacial dos sistemas Unix. Os dois sistemas instalados em cada entreposto foram IBM (modelos S7A e H50) com o sistema operativo AIX. Cada sistema foi instalado numa sala distinta e entre ambos foi definido um cluster de alta disponibilidade em HACMP para AIX.

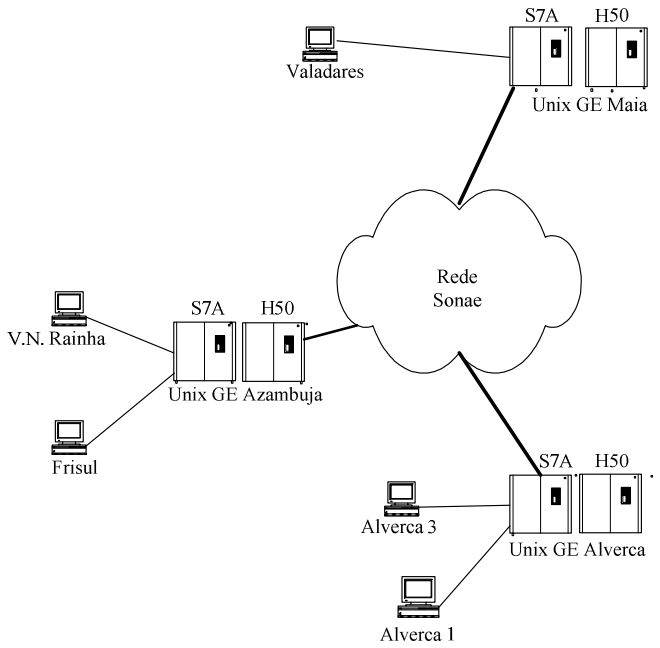


Figura 6-2 – Infra-estrutura de servidores AIX envolvida no projecto NPO-Entrepostos.

A Figura 6-3 ilustra o esquema geral da rede da Sonae Distribuição. Antes do projecto NPO-Entrepostos, o sistema de informação era suportado por servidores IBM AS/400. O objectivo deste projecto consistiu na migração da gestão dos entrepostos para servidores Unix, embora durante algum tempo tenham coexistido as duas infra-estruturas. A gestão das lojas (Modelo e Continente) assentava na aplicação GELO, desenvolvida pela Enabler, também em servidores IBM AS/400. A centralização dos pedidos das lojas, das encomendas aos fornecedores e da gestão dos transportes era assegurada pela direcção comercial através de uma aplicação de Gestão de Inventário. Em termos gerais e com as actualizações tecnológicas verificadas, a plataforma operacional implementada continua a vigorar nos dias de hoje e é muito semelhante a de outras empresas na área da distribuição e retalho.

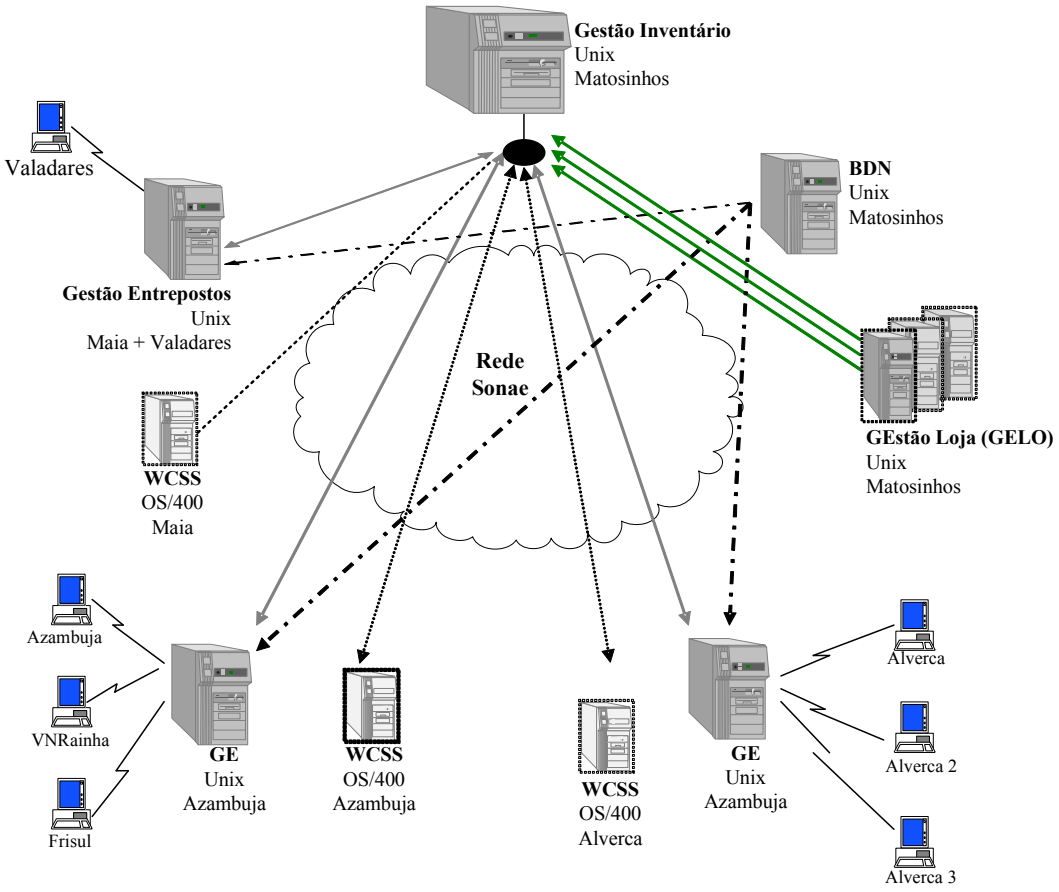


Figura 6-3 – Topologia geral de rede da Sonae Distribuição.

É fácil constatar que o sistema de informação integrado da cadeia de abastecimento assentava fortemente numa infra-estrutura de comunicações que se pretendia segura e fiável. A rede de comunicações assentava na altura num *link* principal entre os edifícios de Matosinhos e Amadora, com uma largura de banda de 2Mbps. Os circuitos de comunicação utilizados na rede eram *frame relay*, com capacidade ajustada às necessidades de cada entreposto. As ligações aos entrepostos principais (Maia, Azambuja e Alverca) eram de 1 Mbps, ligando ao ponto principal mais próximo. No caso concreto do entreposto da Maia, o circuito era Maia-Matosinhos, sendo os restantes estabelecidos à Amadora (Azambuja-Amadora e Alverca-Amadora). Relativamente aos entrepostos de menor dimensão (por exemplo, o de Vila Nova da Rainha), os circuitos tinham uma capacidade na ordem dos 128 Kbps. Todos os circuitos instalados possuíam uma linha alternativa de *backup*, assegurando assim a disponibilidade da ligação à rede pública.

6.2 METODOLOGIA

De uma forma geral, conforme se ilustra na Tabela 6-2, a implementação do projecto em cada um dos entrepostos teve três fases principais: definição do ambiente de produção, testes de integração e, por fim, implementação e *roll out* para produção.

Arquitectura	Testes	Implementação
• Preparar ambiente de teste • Treinar utilizadores finais • Preparar ambiente de produção • Validar arquitectura tecnológica • Definir modelo de suporte	• Testar integração aplicacional • Testar integração tecnológica • Realizar testes de volume	• Instalar e configurar a infra-estrutura tecnológica • Assegurar o suporte técnico ao cliente nas diferentes fases de implementação • Roll-out do processo de migração • Desenvolver automatismos de gestão e monitorização dos servidores do entreposto

Tabela 6-2 – Metodologia adoptada no projecto NPO-Entrepostos.

O ambiente de treino e formação dos utilizadores finais foi o mesmo em todos os entrepostos. Para o efeito foi instalado um sistema AIX de testes com a aplicação de gestão de entrepostos, permitindo testar os processos de integração e migração de dados. Foram igualmente realizados neste ambiente, testes de volume e de robustez da aplicação e das configurações adoptadas para o servidor.

A Figura 6-4 ilustra os servidores Unix utilizados no projecto. Em cada entreposto foi instalado o ambiente de produção. As alterações à aplicação eram realizadas e validadas num servidor destinado para o efeito (*Quality Assurance*, ou QA). Após a validação, essas alterações eram entretanto reflectidas nos ambientes de treino e de produção. O ambiente de treino era utilizado para a formação dos utilizadores dos entrepostos nas novas funcionalidades que iam sendo desenvolvidas.

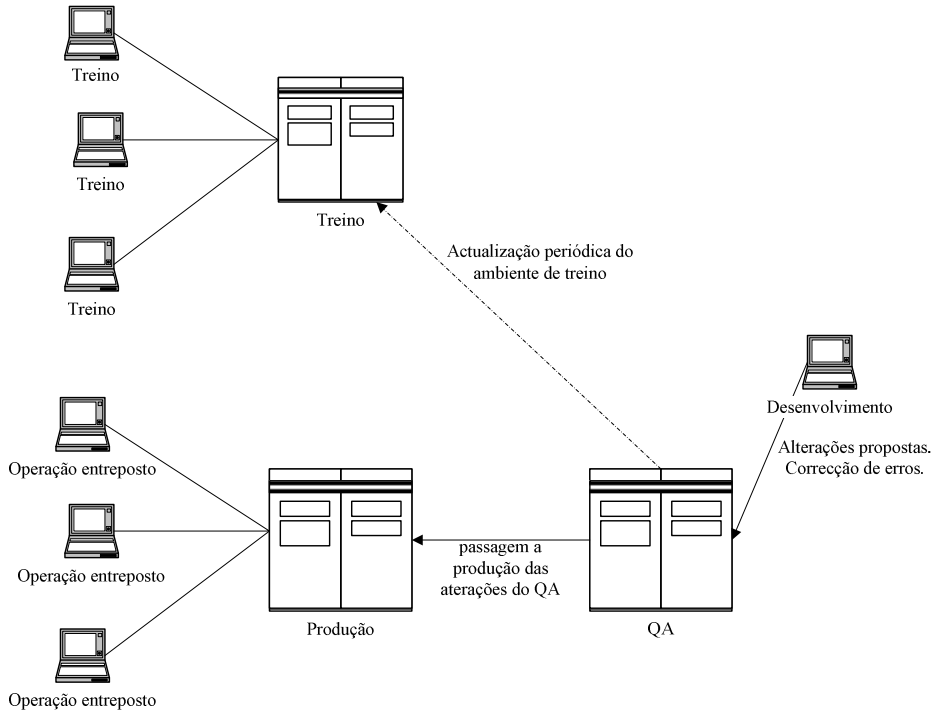


Figura 6-4 – Servidores AIX envolvidos no projecto e correspondente interligação.

6.3 GESTÃO DO PROJECTO

Desde o início do projecto assegurei a função de gestor de projecto na área de implementação e configuração da plataforma tecnológica, representada pelo parceiro SRD. As tarefas desempenhadas durante a fase de projecto, em cada entreposto, incidiram nas seguintes áreas:

- **Sistemas Unix de teste, QA e produção.** Dada a experiência em sistemas Unix, assumi a instalação e configuração desta componente tecnológica, nomeadamente o sistema operativo AIX, a base de dados, o HACMP e o ADSM para gestão centralizada de *backups* (Capítulo 7). Foram igualmente assumidas todas as tarefas de gestão e administração dos servidores Unix durante as fases de testes e arranque em cada um dos entrepostos.
- **Rede de rádio frequência.** Coordenação dos trabalhos de instalação, configuração e testes da rede de rádio frequência em todos os entrepostos, bem como de todos os terminais portáteis.
- **Arquitectura de contingência do projecto.** Elaboração do plano de contingência do projecto. Dado o elevado grau de risco do projecto, bem como o impacto no negócio, foi definida uma arquitectura de contingência a vários níveis, descrita mais adiante neste capítulo.
- **Modelo de suporte.** Definição, juntamente com os restantes parceiros do projecto, do modelo de suporte a ser implementado no âmbito do projecto NPO - Entrepostos. Esse modelo foi definido genericamente para todos os entrepostos e envolveu a Modis, a SRD e a Enabler. O suporte ao projecto NPO foi definido numa base 24x7 nas componentes aplicacional e tecnológica. A Modis definiu os níveis de serviço exigidos ao qual o modelo de suporte deveria corresponder.
- **Ferramentas de monitorização**. Análise, desenho e colaboração no desenvolvimento de aplicações de monitorização das infra-estruturas aplicacional e tecnológica do projecto. As aplicações desenvolvidas foram integradas nos processos de monitorização da equipa de operações da SRD, tendo-se definido procedimentos de actuação nos diversos tipos de alerta possíveis.
- **Definição de política de *backups*.** Definição da política de *backups*, englobando os esquemas de rotatividade, o planos de *disaster recovery* e os procedimentos de *backup* diário.
- **Formação**. Planeamento e coordenação de várias acções de formação às equipas de suporte e operação. Foram efectuadas acções de formação sobre os componentes tecnológicos do projecto (manipulação, configuração e tarefas de *troubleshooting*), as ferramentas de monitorização desenvolvidas e ainda sobre os fluxos principais existentes entre a aplicação de gestão de entreposto e outras existentes no sistema de informação, nomeadamente a das lojas e gestão de inventário (Secção 6.6).
- **Pré-venda:** Foram realizadas várias apresentações internas sobre o projecto, nomeadamente, o seu estado actual, evoluções futuras, principais problemas encontrados e

metas atingidas. Realizei ainda várias apresentações técnicas sobre o projecto relativas aos aspectos tecnológicos envolvidos.

- **Documentação**. Elaborei vários documentos técnicos, sintetizados em [48], com os procedimentos desenvolvidos no projecto, onde se destacam os seguintes:
    - Acções de gestão do sistema Unix.
    - Manuais de utilização e configuração dos componentes tecnológicos.
    - Propostas para a arquitectura de alta disponibilidade, evidenciando os vários cenários e o grau de risco associado a cada um.
    - Política de *backups* definida para o projecto e respectivo manual de operação.
- **Consultoria técnica.** Relativamente à componente tecnológica do projecto, produzi algumas propostas técnicas:
    - Auditoria de alguns componentes do projecto e a sua vulnerabilidade a eventuais falhas de segurança.
    - Proposta de adequação de alguns processos de negócio à nova infra-estrutura, na tentativa de tirar partido dos componentes tecnológicos envolvidos e consequentemente na melhoria do desempenho global da solução.

6.4 CONSTRANGIMENTOS

Este projecto estava, logo à partida, revestido de vários aspectos que o tornavam bastante crítico para o negócio da Sonae Distribuição em geral, e para as operações de gestão de entreposto e de inventário, em particular. Os principais pontos críticos identificados foram:

- **O modelo de negócio.** O negócio do retalho é, por definição, bastante crítico, implicando que uma alteração no sistema de informação da gestão de entrepostos tenha repercussões (directas ou indirectas) com os restantes componentes da cadeia de distribuição, nomeadamente as lojas e os fornecedores. Este constrangimento foi visível, entre outros aspectos, na janela de tempo apertada para intervenções na infra-estrutura tecnológica e pela necessidade de assegurar mecanismos de transição nas aplicações e na infra-estrutura de rede e servidores utilizada.
- **"Bug" do ano 2000:** A aplicação de gestão de entrepostos que existia na altura não tinha corrigido o problema do formato da data, conhecido como o "bug" do ano 2000. Era um problema já identificado há bastante tempo e foi assumido que não se resolveria, já que a migração para uma nova aplicação de gestão de entrepostos seria um facto antes da entrada no ano 2000.
- **Tecnologia associada ao projecto.** As alterações provocadas pela nova aplicação de gestão de entrepostos envolviam tecnologia nova, não só para os operadores do entreposto mas também para as equipas técnicas de informática.
- **Período de transição com duas aplicações distintas.** A impossibilidade de migrar simultaneamente para o novo sistema em todos os entrepostos obrigou à manutenção das

duas aplicações em simultâneo. Este facto obrigou à implementação de várias interfaces aplicacionais entre os dois sistemas de gestão do entreposto.

- **Complexidade do pacote aplicacional adquirido**. Embora fosse bastante versátil e flexível, o grau de complexidade aumentou substancialmente, dando origem a alguns constrangimentos, especialmente durante a fase de parametrização.

6.5 FUNDAMENTOS TEÓRICOS

Esta secção contempla uma descrição sumária dos principais conceitos associados à distribuição, à logística e ao inventário, bem como à gestão de operações num entreposto.

O negócio do retalho, tanto alimentar, como não alimentar ou sazonal, trabalha com artigos de consumo vendidos nos hipermercados e noutras lojas especializadas. A cadeia de distribuição desses artigos tem início nos centros de produção, onde são fabricados, só terminando nos hipermercados onde são postos à disposição do consumidor final. Pelo caminho a maioria dos produtos passou por um entreposto. A Figura 6-5 ilustra o circuito percorrido pelos produtos desde o fabrico até ao consumidor final.

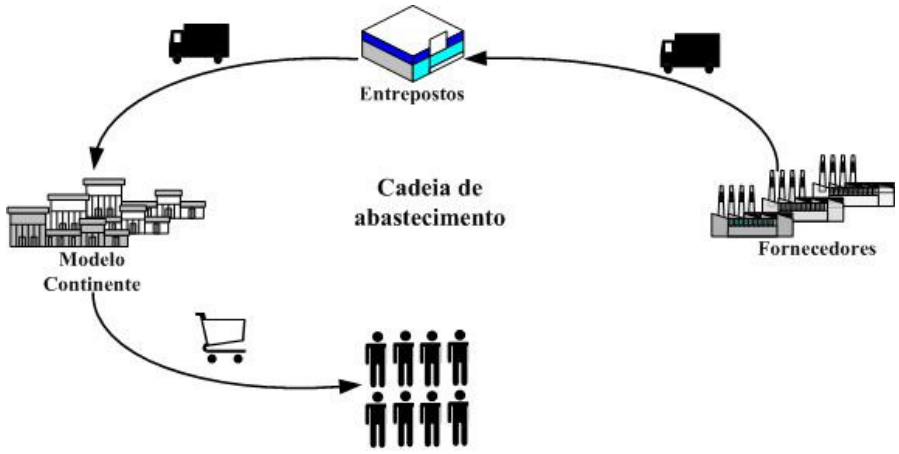


Figura 6-5 – Visão global da cadeia de abastecimento.

6.5.1 GESTÃO DE INVENTÁRIO

Em empresas de retalho de menor dimensão é possível que a gestão de inventário esteja localizada directamente nas lojas, ficando cada uma responsável pela negociação, colocação de encomendas no fornecedor e gestão do seu *stock*, com o máximo de rentabilidade e um risco mínimo de quebras. Por quebra entende-se a destruição física das mercadorias devido a roubos, incêndios ou mau manuseamento. Por seu lado, a ruptura de *stock* corresponde à não satisfação da procura por parte dos clientes finais devido à falta de mercadoria em *stock*.

Com o surgimento das cadeias de lojas passou-se a racionalizar os meios necessários à gestão dos artigos postos à disposição dos clientes, com tendência para a centralização dessa função num único ponto, o entreposto, onde os produtos são concentrados antes de serem expedidos para os pontos de venda. Por este motivo, as lojas passaram gradualmente a ser abastecidas por um único fornecedor, o entreposto. Mesmo nas excepções, em que o fornecedor entrega a mercadoria

directamente às lojas, mantém-se a condição de ser a direcção do entreposto a realizar a encomenda ao fornecedor, acordando a entrega directamente na loja. Esta estratégia trás vários benefícios para as cadeias de lojas, destacando-se a redução de custos provocada pelo factor escala, a redução de custos fixos de operação e manutenção da mercadoria e a diminuição da área de armazenamento necessária em cada loja.

Por Gestão de Inventário (GI) [70-73] entende-se a função que pretende optimizar os níveis de *stock* e de ruptura na cadeia, minimizando os custos de armazenamento e aumentando a rotação dos artigos. É ainda responsável pela coordenação entre a operação do entreposto e os fornecedores, através da mediação de uma direcção comercial. Os sistemas de GI recorrem a um conjunto de regras e procedimentos que os transformam num sistema de apoio à decisão, tomando decisões automáticas baseadas na informação disponível.

6.5.2 GESTÃO DE ENTREPOSTO

A operação de logística [73, 74], integrado na gestão do entreposto (GE) comporta as actividades de movimentação e armazenamento de mercadorias nos entrepostos e destes às lojas, com um nível de serviço adequado ao cliente e um custo mínimo associado. As actividades centrais da actividade logística são:

- Transporte de mercadorias entre os entrepostos e as lojas.
- Gestão do *stock* com vista a alcançar um nível de disponibilidade dos produtos nas lojas adequado às necessidades dos consumidores finais.
- Processamento das encomendas de cada loja e envio das respectivas notas aos fornecedores.

A gestão de *stocks* configura-se como o ponto de maior afinidade entre a logística e a GI. Estas três actividades, para além da sua importância estratégica para a operação, também são consideradas essenciais por contribuírem com a maior parcela dos custos da actividade logística, e consequentemente contêm um elevado potencial de optimização. Há ainda duas actividades importantes da responsabilidade directa da operação do entreposto: armazenamento (gestão de espaço) e manuseamento (movimentação de produtos).

Em termos organizacionais, a logística tem vindo a optimizar os seus serviços em duas vertentes principais: operacional, relativo à utilização de tecnologias mais eficientes nos equipamentos utilizados na operação do entreposto; de gestão, relacionado com a negociação de acordos de parceria e colaboração entre os entrepostos e os fornecedores.

6.5.3 MOVIMENTAÇÃO DE PRODUTOS NO ENTREPOSTO

A movimentação de mercadorias na cadeia de distribuição implica o manuseamento do produto por diversas vezes ao longo o seu percurso. Concretamente no entreposto, o objectivo é a movimentação rápida, e a baixo custo, das mercadorias. Para tal, é possível encontrar no entreposto uma grande variedade de equipamentos mecânicos destinados ao manuseamento de um número variado de produtos de diversos tamanhos, formas, volumes e peso. A Figura 6-6 ilustra alguns desses equipamentos e alguns esquemas de acondicionamento dos produtos no entreposto.

O manuseamento eficiente das mercadorias dentro do armazém depende consideravelmente do projecto de estruturação, disposição e organização do espaço. Pequenas variações na configuração, organização física e localização dos cais de carga e expedição podem traduzir-se em ganhos consideráveis de eficiência.


www.forkliftwestsussex.com


www.directindustry.com


www.m-h-e.com

Figura 6-6 – Exemplos de equipamentos de transporte e armazenamento.

6.5.4 CODIFICAÇÃO E IDENTIFICAÇÃO DOS PRODUTOS

Para facilitar a sua identificação em todo o sistema de informação do negócio, cada produto tem um código único de identificação, que se encontra gravado num ou em vários lados da sua embalagem. Este código é gravado sob a forma de código de barras utilizando a notação EAN[7]. Esta norma de codificação de artigos é também usada nas frentes de caixa dos supermercados, facilitando as operações de actualização de *stock* do produto.

6.5.5 EFFICIENT CONSUMER RESPONSE

O Efficient Consumer Response (ECR) consiste na definição de parcerias de negócio entre distribuidores e fornecedores, fomentadas pelo mercado, tendo como consequência principal a maior satisfação do consumidor e a redução dos custos. Neste sistema, tanto os produtos como a informação fluem em tempo real, desde a linha de produção até às frentes de caixa dos hipermercados através de canais que dispensam a utilização de papel. A informação trocada entre os parceiros comerciais é, portanto processada de uma forma rápida, transparente e integrada. Para o efeito muito contribuíram os sistemas de EDI, como o projecto TEDI descrito no Capítulo 4, que permitiu agilizar a troca de informação electrónica entre os fornecedores, o entreposto e as lojas.

Há quatro conceitos importantes implementados num sistema ECR que se descrevem de seguida: Vendor Management Inventory, Pick-by-Line, Cross-Docking e Roll-Cage Sequencing.

A. VENDOR MANAGED INVENTORY

O reaprovisionamento contínuo consiste em procurar atingir, ao longo do tempo, um fluxo homogéneo de produtos entre o fornecedor e o entreposto. O fornecedor gera as encomendas de reaprovisionamento dos entrepostos segundo parâmetros pré-acordados de níveis de serviço,

[7] EAN - European Article Numbering

baseadas na informação de promoções e de *stock* que o distribuidor (direcção comercial) lhe transmite. Posteriormente transmite-a aos entrepostos ou directamente às lojas, para validação. Esta estratégia permite uma diminuição do *stock* de segurança no entreposto, com menor risco de quebra de abastecimento. Em qualquer dos casos, o fornecedor, estando na posse dos planos de encomendas dos seus clientes, pode planear de forma mais segura a sua produção. Outra das vantagens deste programa consiste em aumentar significativamente os indicadores do nível de serviço do entreposto.

B. PICK-BY-LINE

No *Pick by Line* (PBL) o fornecedor entrega os produtos nas quantidades encomendados pelo entreposto. Este efectua a recepção e, de imediato, procede à produção de *paletes* para as lojas, de forma a satisfazer as encomendas dos artigos recepcionados. O *stock* gerado pelo PBL é praticamente inexistente. Com efeito, as quantidades recepcionadas satisfazem normalmente as encomendas das lojas para os produtos em causa, que são expedidas logo após serem recepcionadas no entreposto.

No entanto, poderá ocorrer *stock* quando as entregas excedem os valores encomendados pelas lojas. Este excesso não provoca aprovisionamento, ficando estes produtos numa área temporária no armazém. No dia seguinte, os produtos que sobraram do PBL do dia anterior são os primeiros a ser expedidos. Os produtos envolvidos no PBL são normalmente os perecíveis, tais como, ovos, leite e legumes, cujo armazenamento prolongado se tornaria inviável. A Figura 6-7 [69] ilustra os principais processos envolvidos na operação de PBL, muito utilizada nos produtos perecíveis que chegam diariamente às lojas provenientes dos entrepostos alimentares.

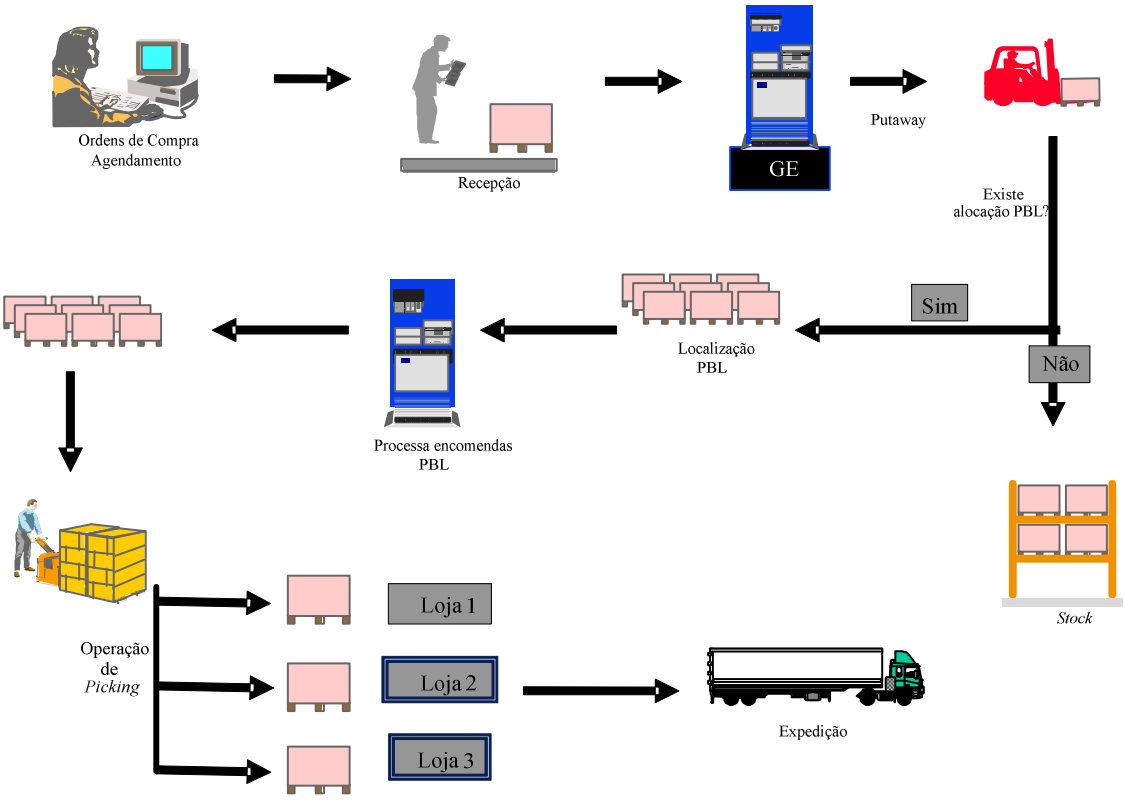


Figura 6-7 – Operação de *pick-by-line*.

C. CROSS-DOCKING

O *cross-docking* consiste na utilização do armazém apenas como uma plataforma de transferência de mercadoria, não havendo necessidade de armazenamento. Os produtos são entregues no entreposto pelo fornecedor, e procede-se ao seu transbordo e distribuição por vários camiões que, por sua vez, irão abastecer as lojas. Desta forma reduzem-se custos operacionais e de aluguer de espaço de armazenamento. No *cross docking* o fornecedor entrega as *paletes* com os produtos encomendados por loja e cada *palete* pode conter mais do que um produto do mesmo fornecedor.

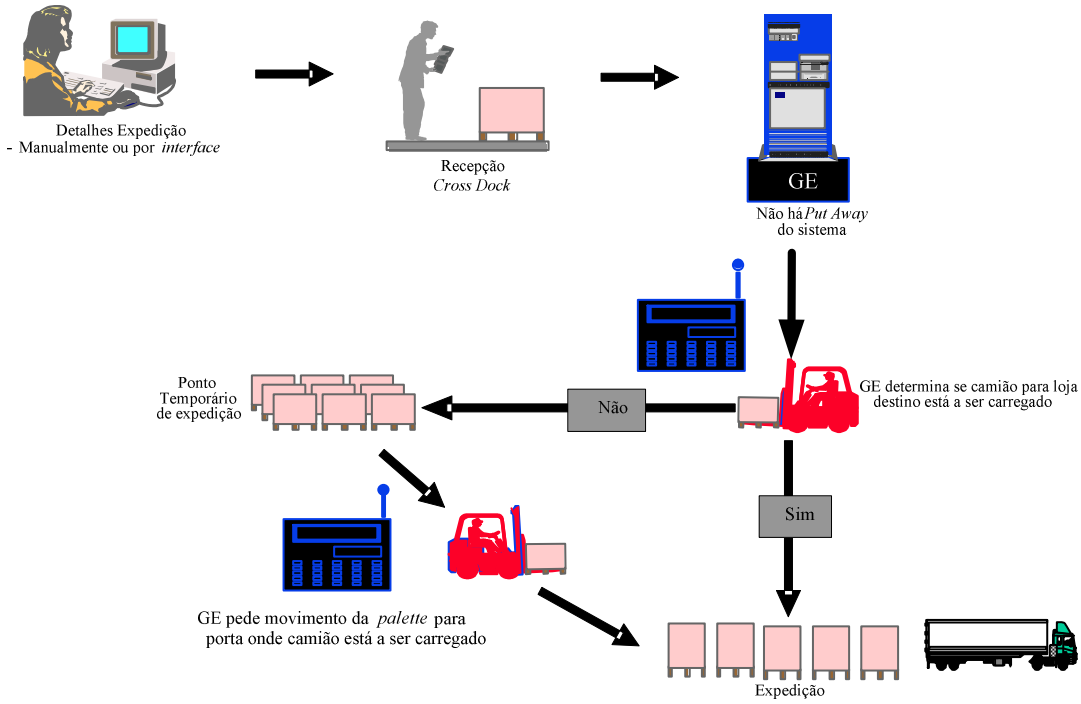


Figura 6-8 – Operação de *cross docking*.

A Figura 6-8 [69] ilustra o processo de *cross-docking* que, a par do PBL representa um dos mais importantes e comuns na gestão do entreposto.

D. ROLL-CAGE-SEQUENCING

Esta técnica pretende assegurar que a localização dos produtos no armazém se organiza, tanto quanto possível, de forma a facilitar a recolha da mercadoria durante a preparação das encomendas para as lojas. É conseguida, nomeadamente através da proximidade física dos produtos de um mesmo grupo, com vista à diminuição do tempo de preparação. O *layout* do armazém terá, assim, de ser pensado de modo a que a preparação das cargas de expedição para as lojas se encontrem todas no mesmo grupo de prateleiras.

Uma variante desta estratégia consiste na reserva de um espaço no chão para cada loja. Neste caso, a colocação da mercadoria na área destinada a cada loja é feita à medida que os produtos vão sendo entregues pelo fornecedor.

Há uma área no entreposto, denominada de *twilight*, que está normalmente reservada a produtos que foram rejeitados durante a fase de preparação de encomendas. Esta rejeição pode ocorrer devido à lotação da área de operação junto aos cais (área de *picking*), ou devido a produtos cuja localização física no entreposto não existe.

6.5.6 SISTEMAS DE REAPROVISIONAMENTO E PREVISÃO

Por reaprovisionamento entende-se o reabastecimento dos armazéns, para que o nível de *stock* seja suficiente para a satisfação de todas as encomendas das lojas. Desta forma, os entrepostos funcionam como um *buffer* entre os fornecedores e as lojas, que se destina a amortecer qualquer quebra na cadeia, como por exemplo, falha de produção, entrega tardia dos fornecedores, ou procura excepcional de certos produtos nas lojas. O reaprovisionamento pode ser optimizado através de algumas ferramentas, como a previsão da procura baseada nas vendas em períodos anteriores, a antecipação de compras com base em informações de quebra de fornecimento, o aumento de preços e o número de vezes que as encomendas não são satisfeitas na totalidade.

6.6 SISTEMA DE INFORMAÇÃO

A nova plataforma operacional de gestão dos entrepostos da Sonae Distribuição, cuja instalação teve início sensivelmente em 1998, surgiu da necessidade de implementar algumas funcionalidades aplicacionais até à altura inexistentes, suportada numa plataforma tecnológica robusta e moderna. A aplicação adquirida contemplou apenas a componente de GE, nomeadamente a recepção de mercadorias, o aprovisionamento (ou *put away*), o reaprovisionamento automático, a preparação e expedição de mercadorias, o PBL, o *cross docking* e o entrelaçamento (*interleaving*) de operações escalonadas para os terminais portáteis dos operadores do entreposto (*pickers*).

Relativamente ao módulo de GI, centralizava um conjunto importante de informação que permitia aos gestores de inventário ter, em cada momento, uma visão concreta do negócio, nomeadamente o *stock* existente e as necessidades de compra para evitar a quebra nas lojas. As suas principais funções consistiam em:

- Centralizar a informação relativa a artigos e fornecedores.
- Centralizar a informação relativa à gestão de *stocks* dos artigos.
- Difundir para os entrepostos as necessidades de expedição de cada produto, com base nas encomendas efectuadas pelas lojas da cadeia.
- Difundir para as lojas (*Backoffice*) as facturas relativas à expedição;

A Base de Dados de Negócio (BDN) centralizava a informação relativa a artigos e fornecedores e difundia-a para as aplicações *datawarehouse*, contabilidade, EDI, Gestão de Lojas (GELO), Gestão de Entrepostos (GE) e Gestão de Inventário (GI). A informação da BDN era actualizada pelos departamentos de gestão da loja, registando toda a actividade do negócio. A aplicação GELO enviava periodicamente os dados para a BDN, alojada na altura nos servidores centrais.

A Figura 6-9 ilustra de forma macro a arquitectura do sistema de informação da Sonae Distribuição e o enquadramento dos módulos de gestão de entrepostos e de inventário. Na figura é visível o carácter central do módulo GI e a difusão de informação do negócio a partir da BDN para todos os módulos.

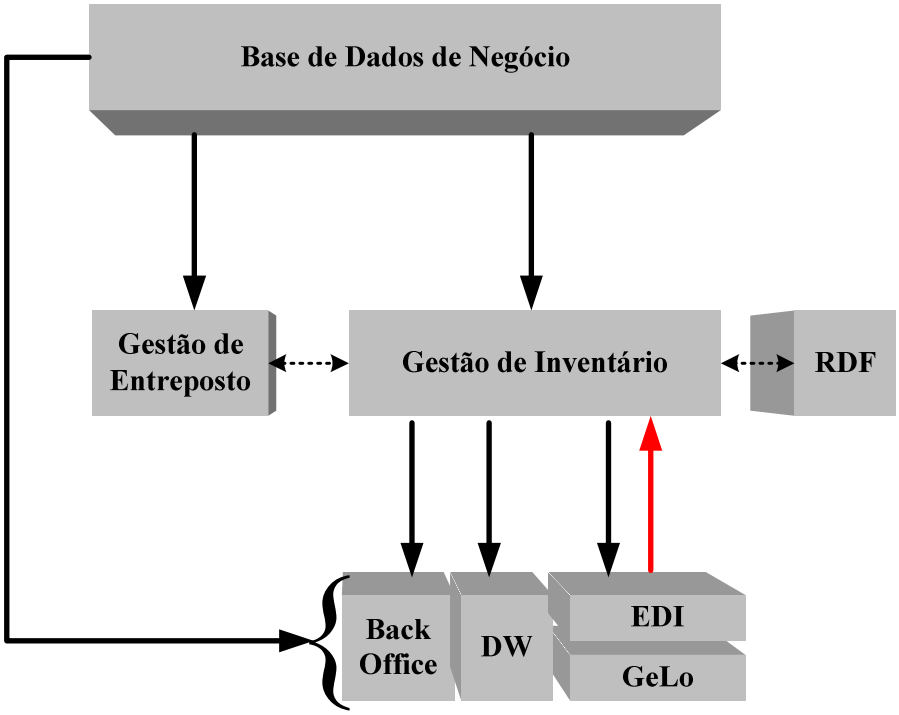


Figura 6-9 – Fluxo de dados no sistema de informação da Sonae Distribuição.

A estrutura apresentada ainda se mantém actual, embora as tecnologias de TI e SI envolvidas tenham sido naturalmente actualizadas. Os fluxos de dados entre as aplicações BDN, GI e GE estão ilustrados na Figura 6-10.

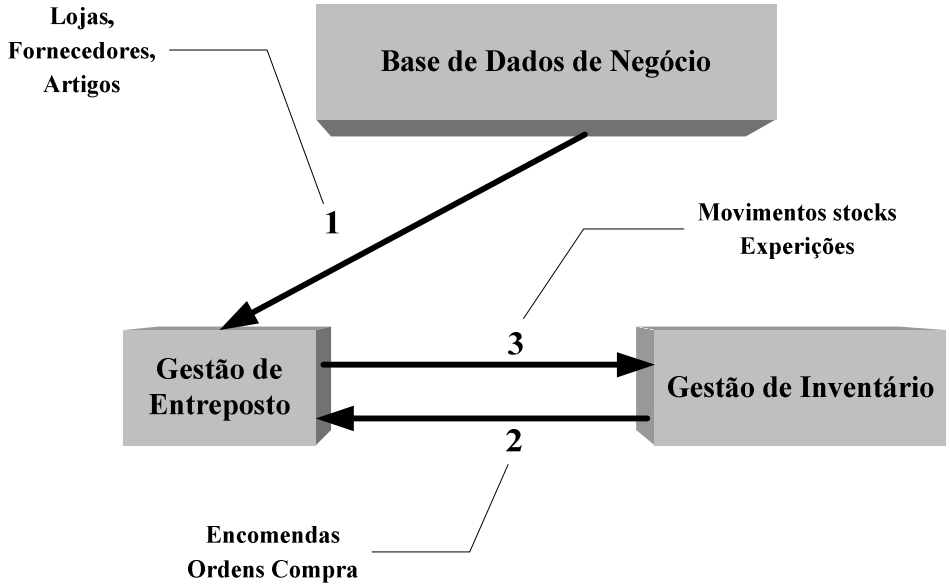


Figura 6-10 – Fluxos de dados entre a BDN, GI e GE.

Os fluxos mais importantes que afectam directamente a GE são os seguintes:

- **Fluxo 1** - relativo ao envio de dados da BDN para a GE. Além deste fluxo para o entreposto é igualmente efectuado um outro para a GI.
- **Fluxo 2** - corresponde ao envio, pela GI, das encomendas normais e de PBL, bem como as ordens de compra aos fornecedores. Esta informação é enviada da GI para a GE.
- **Fluxo 3** - que engloba os documentos relativos aos movimentos de *stock* e expedições, enviados da GE para a GI e, posteriormente, para as lojas.

O primeiro fluxo ocorria apenas uma vez por dia e correspondia a uma difusão da BDN para todos os componentes descritos, entre os quais a GE e a GI. A GI enviava encomendas (*outbound*) para o entreposto num intervalo de tempo acordado. O envio de dados em sentido inverso, da GE para a GI, ocorria com uma periodicidade maior. Quanto ao terceiro fluxo correspondia ao tempo estimado de deslocação entre o entreposto e a loja mais próxima. Desta forma, a mercadoria era expedida e, quase em simultâneo, era enviada electronicamente a factura e os movimentos de *stock* originados.

6.7 PLATAFORMA TECNOLÓGICA

A infra-estrutura tecnológica do projecto englobou vários componentes. Nesta secção descrevem-se os mais importantes, as suas características e a utilidade no projecto NPO. Na descrição será dado ênfase à arquitectura de sistemas AIX adoptada, nomeadamente, a arquitectura de contingência, utilizando o HACMP, as ferramentas de monitorização desenvolvidas, o modelo de suporte à solução final e o sistema de *backups* centralizado.

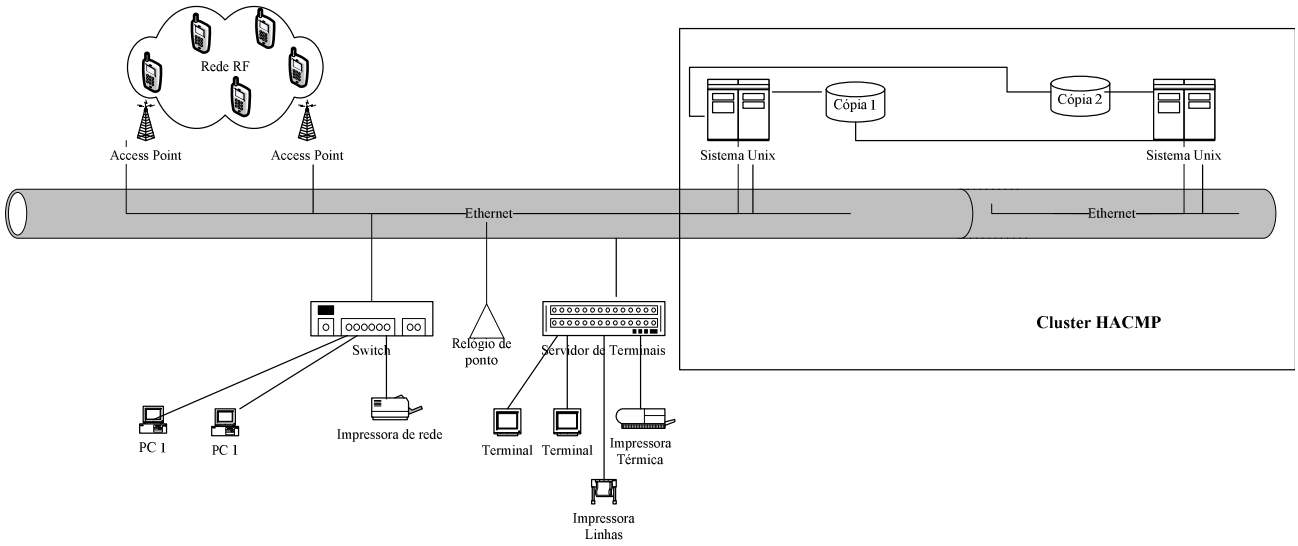


Figura 6-11 – Arquitectura global do projecto NPO.

A Figura 6-11 ilustra a solução global do projecto na sua vertente tecnológica [69]. De seguida descrevem-se sucintamente os principais componentes.

6.7.1 A REDE LOCAL E DE ACESSO REMOTO

A plataforma tecnológica deste projecto assentou na infra-estrutura de rede TCP/IP já existente nos entrepostos. Para este projecto foi realizado o planeamento das necessidades de endereçamento IP para todos os componentes do projecto e configurados os equipamentos em conformidade. Foram igualmente utilizadas as infra-estruturas de acesso à rede *frame relay* a partir de dois pontos principais, Matosinhos e Amadora.

6.7.2 A REDE DE RÁDIO FREQUÊNCIA

A rede de rádio frequência (RF) assentou na infra-estrutura de rede local Ethernet do entreposto. As unidades móveis comunicavam com os pontos de acesso (AP) via rádio. Ao contrário da solução existente, o novo sistema de informação da GE contemplava o uso de dispositivos móveis, destacando-se as seguintes vantagens principais:

- Possibilitar o acesso rápido à informação a partir de qualquer ponto do entreposto.
- Reduzir o tempo de localização dos produtos no entreposto e da manutenção do seu *stock*, garantindo maior rotatividade e precisão de localização.
- Reduzir os custos operacionais.
- Reduzir o número de operacionais necessários para realizar uma transacção completa de dados.

Os componentes da rede de rádio frequência adquiridos para cada entreposto contemplaram:

- Terminais portáteis para a recolha de dados.
- AP para receber e transmitir informação entre os terminais portáteis e o sistema central. Os AP foram agrupados fisicamente de forma a criarem uma estrutura celular de cobertura tridimensional da área.

Uma vez que a potência dos rádios incorporados nos AP era, em certa medida reduzida, foi determinante a colocação estratégica dos AP, de forma a garantir uma cobertura óptima do local de operação dos terminais. Assim, foi conduzido um estudo preliminar de avaliação do número e localização dos AP, tendo em vista a cobertura óptima.

Os terminais portáteis de rádio frequência constituem um dos extremos da cadeia de equipamentos que integram a solução de RF implementada. Estes terminais são utilizados pelos operadores durante a execução das suas tarefas no terreno. A sua missão é a de servir de ponto de acesso ao sistema central de processamento, quer através de uma emulação de terminal quer pela execução de uma aplicação específica, como a leitura de código de barras dos produtos. Dessa forma tornou-se possível a visualização da informação sobre o produto em tempo real, possibilitando a actualização automática das bases de dados. Por exemplo, a actualização automática do *stock* de um determinado produto após a sua recepção ou expedição.

Neste projecto foram utilizados dois tipos de equipamentos portáteis; os *handheld*, ou "de mão", utilizados pelos operadores apeados para efectuar as tarefas de recepção, e os *forklift*,

instalados nos empilhadores, para as tarefas de preparação de encomendas, reaprovisionamento e expedição.

Anteriormente, as tarefas eram atribuídas aos operadores do entreposto através de mapas em papel. Com a nova plataforma operacional estes mapas foram substituídos pelos terminais de rádio frequência, onde as tarefas a realizar apareciam de forma automática.

6.7.3 O SISTEMA UNIX

O sistema central, em cada entreposto, era constituído por dois sistemas IBM RISC/6000, um do modelo S7A e outro do modelo H50. Estes dois sistemas formavam um cluster de HACMP, conforme descrito na Secção 6.10.4. Para realizar os *backups* ao sistema foi usada a aplicação ADSM, detalhada no Capítulo 7. Em termos de *hardware* dedicado para *backups*, foram instaladas duas unidades de *backup* (usando *tapes*). Uma das unidades era interna a cada sistema e a outra, externa, era do modelo IBM 7337 e partilhada pelas duas máquinas. Esta última dispunha de duas unidades de leitura/escrita e um *slot* de 10 *tapes* com expansão para mais 5. As *tapes* utilizadas tinham capacidade para 70Gbytes com compressão de dados.

Os dois discos internos com o sistema operativo AIX foram interligados pelo protocolo SCSI. Quanto aos discos externos, foram configurados em *mirror*, através do protocolo SSA. A solução adoptada relativamente aos discos partilhados comportou *mirror* dos discos e duplicação das ligações SSA. Foi igualmente implementada uma solução de *stripping* com vista ao aumento do desempenho de acesso aos discos. Através desta técnica foi possível distribuir as partições lógicas dedicadas às aplicações pelos vários discos físicos disponíveis. Assim, cada partição lógica existia fisicamente em vários discos e não apenas num. Em suma, aliando as técnicas de *mirroring* e *stripping* na configuração dos discos externos tornou-se possível obter melhor desempenho no acesso aos dados e aumentar o nível de disponibilidade dos discos. A Figura 6-12 ilustra os aspectos descritos anteriormente sobre a interligação dos discos e unidades de *backup* aos sistemas Unix.

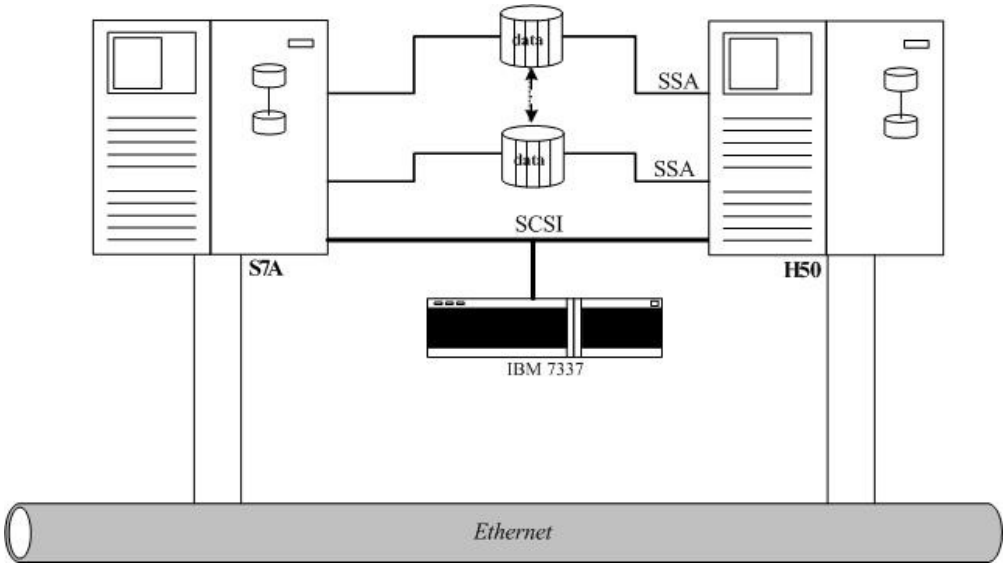


Figura 6-12 – Interligação das unidades de armazenamento de dados com o servidor Unix.

6.7.4 OUTROS EQUIPAMENTOS

Pela sua natureza, o projecto utilizou equipamentos específicos da operação dos entrepostos, destacando-se as impressoras térmicas e os relógios de ponto.

A. IMPRESSORAS TÉRMICAS

As impressoras utilizadas para imprimir as etiquetas na preparação das encomendas eram do tipo térmico e foram localizadas nas áreas de recepção e expedição do entreposto. Para aumentar a disponibilidade de algumas destas impressoras, foi utilizado um dispositivo que as ligava igualmente à rede de rádio frequência.

B. RELÓGIOS DE PONTO

A aplicação adoptada contemplava um módulo de controlo da actividade dos operadores do entreposto. Sempre que o operador do entreposto iniciava o seu turno ou o período de descanso, essa informação era registada através de relógios de ponto com um leitor de código de barras, instalados estrategicamente no entreposto. Após a sua entrada no turno eram-lhe imediatamente atribuídas tarefas de operação no entreposto. Cada relógio encontrava-se ligado por rede ao sistema central e, sempre que solicitado, recolhia o número de colaborador (através da leitura do código de barras) e o tipo de evento associado. Estes terminais foram ligados a um *switch* Ethernet, embora pudessem também fazer parte da rede de rádio frequência instalada.

6.8 APLICAÇÕES DE MONITORIZAÇÃO

Um dos objectivos principais da equipa de projecto da SRD consistia na elaboração de um manual de procedimentos coerente e optimizado para ser implementado pela equipa de suporte, após o *roll out* e período de estabilização em cada entreposto. Foram as seguintes as aplicações de monitorização desenvolvidas:

- **Monitorização de fluxos**: Monitorização dos fluxos descritos anteriormente (Secção 6.6), destacando-se as seguintes características principais:
    - Multi-plataforma.
    - Utilizava *checkpoints* para retomar transmissões interrompidas a partir do momento em que ocorreu o erro.
    - Possuía uma *interface* gráfica para gestão dos processos de monitorização existentes.
    - Registava os eventos associados a cada processo de monitorização dos fluxos.
- **Monitorização das comunicações:** Integração da monitorização das comunicações envolvidas no projecto (entre os entrepostos e destes com a direcção de loja e gestão de inventário) na plataforma de monitorização da SRD.

- **Monitorização do sistema Unix:** Foi desenvolvido o Sistema de OPerações ASsitido (SOPAS), que disponibilizou aos operadores de sistemas Unix da SRD um conjunto de utilitários para a monitorização de componentes vitais dos sistemas envolvidos no projecto.

A aplicação SOPAS foi desenvolvida em Unix *Shell Script* e encontrava-se dividida em duas partes principais: monitorização e operação. A tarefa de monitorização dos sistemas encontrava-se escalonada na `cron` para ser executada a cada 15 minutos, verificando o estado de diversos itens:

- Ocupação de *filesystems*.
- Ocupação da área de *swap*.
- Operacionalidade da aplicação e de alguns dos seus componentes.
- Monitorização do desempenho do sistema (processos em fila de execução, nível de paginação, entre outros).
- Teste de comunicação TCP/IP com os restantes sistemas envolvidos no projecto (Figura 6-3).
- Análise de erros de *hardware* do sistema.

Este processo de monitorização era executado em cada sistema Unix e monitorizava os mesmos itens. No entanto, a visualização dos alertas era centralizada no sistema do entreposto da Maia. Para o efeito, o processo do sistema da Maia efectuava uma cópia do ficheiro de monitorização parcial de cada um dos outros dois sistemas (Azambuja e Alverca). Esta medida permitiu a centralização, num só sistema, dos erros e alertas relativos a todos os sistemas envolvidos no projecto. As equipas de operação podiam ainda aceder directamente a uma *shell* de AIX e executar procedimentos simples, de uma forma assistida, relacionados com algumas tarefas de gestão e operação do sistema, ou ainda executar comandos Unix.

Foi ainda desenvolvido um gestor de transacções integrado (GTI), que realizou a gestão, monitorização e registo de todos os fluxos existentes no âmbito do projecto NPO – Entrepostos (Figura 6-9 e Figura 6-10). Este componente de monitorização alertava para os erros ocorridos nas diversas etapas de cada fluxo, indicando eventuais causas e propondo acções para a resolução do problema. A Figura 6-13 apresenta a arquitectura da aplicação GTI, desenvolvida para apoio à gestão e monitorização dos fluxos de dados envolvidos [69].

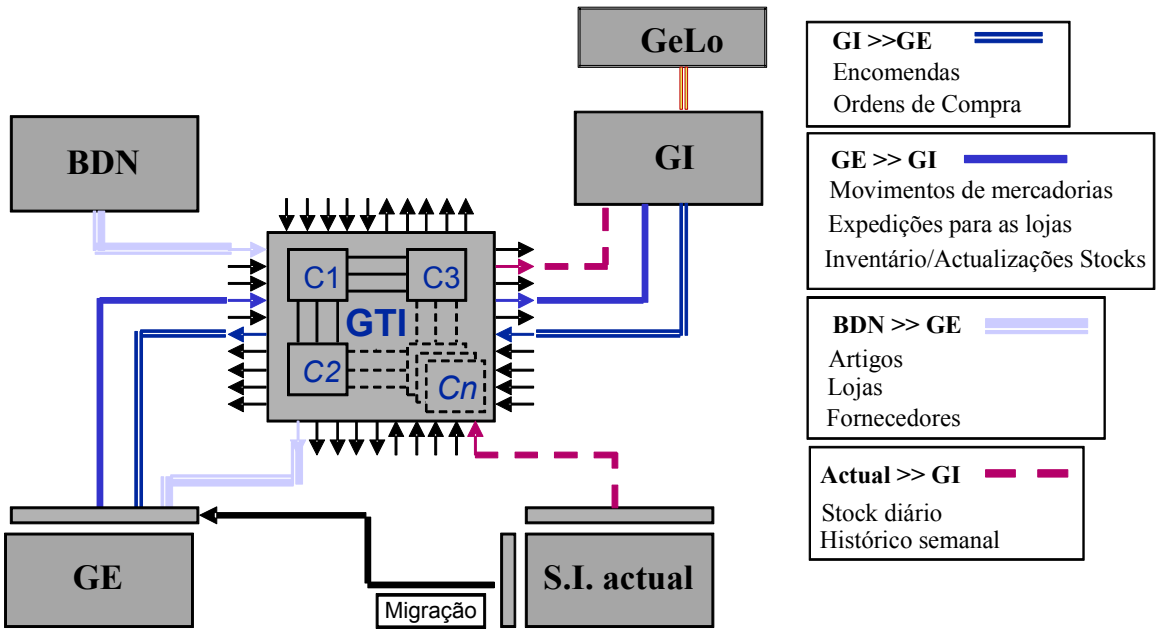


Figura 6-13 – Arquitectura geral da aplicação de monitorização GTI.

A monitorização dos fluxos através do GTI e dos alertas do sistema Unix através do SOPAS constituíram as duas principais aplicações de gestão desenvolvidas para o projecto.

6.9 MODELO DE SUPORTE

Este projecto foi, desde a primeira hora considerado crítico e, como tal, requereu um modelo de suporte objectivo e real que permitisse uma resposta eficiente e eficaz aos alertas registados nos sistemas de cada entreposto. Dessa forma, foram definidas três equipas distintas de suporte, conforme se ilustra na Figura 6-14.

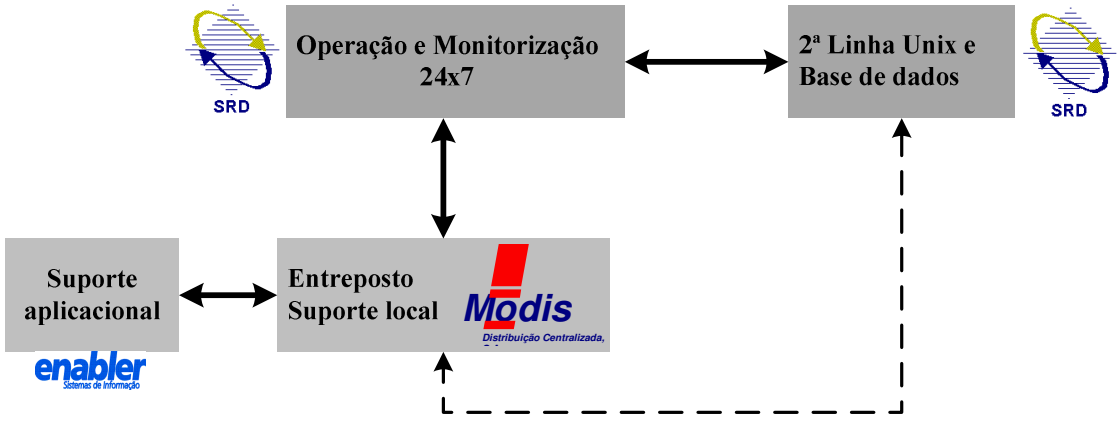


Figura 6-14 – Modelo de suporte do projecto NPO-Entrepostos.

O relacionamento entre os vários grupos de suporte afectos ao projecto e respectivas competências eram os seguintes:

- **Suporte local nos entrepostos**. Constituído por elementos de 1ª e 2ª linha, geograficamente próximos dos entrepostos. Algumas das suas responsabilidades incluíam

a substituição de equipamentos avariados onde existiam outros suplentes, apoio local aos utilizadores e a análise e correcção de erros registados nos fluxos de dados entre a GI e a GE.

- **Operações**. Funcionava numa base de 24x7 a partir do centro de informática da SRD. As suas responsabilidades incluíam a análise dos alertas registados pelas aplicações de monitorização, a tentativa de resolver de imediato o problema e, se não fosse o caso, escalar a 2ª linha de suporte respectiva.
- **2ª Linha de suporte Unix**. Intervinha em horário normal e tinha a responsabilidade de resolver os problemas mais delicados de gestão Unix, não solucionados pela primeira linha de suporte.
- **Suporte remoto à aplicação**. Suporte apenas à aplicação de GE, igualmente numa base de 24x7, da responsabilidade da Enabler.

6.10 ARQUITECTURA DE CONTINGÊNCIA

Atendendo à elevada importância do negócio e deste projecto em particular, houve particular atenção na implementação de arquitecturas de contingência a vários níveis, como sejam instalações físicas, componentes tecnológicos do projecto, a plataforma Unix e o sistema de *backups*. Em termos práticos, pretendeu-se que o sistema estivesse disponível durante o período de actividade dos entrepostos, ou seja, genericamente de 24 horas por dia, durante os sete dias da semana, ao longo de todo ano. Só assim seria possível satisfazer as encomendas das lojas nos prazos considerados, não originando quebra de *stocks*.

Relativamente à plataforma tecnológica, foram implementados alguns procedimentos com vista à mitigação dos pontos críticos de falha e do tempo de reparação em caso de falha. Assim, todos os componentes tecnológicos do projecto tinham uma unidade suplente, pronta e configurada para ser usada em caso de uma avaria. Esses componentes suplentes eram testados periodicamente a fim de garantir que, sempre que fosse necessária a sua participação, se encontrariam operacionais. De acordo com o contracto de manutenção definido com os fornecedores, quando ocorria uma avaria num componente, por exemplo num terminal RF, a substituição era imediata pelo equipamento suplente e a reparação do avariado tinha um período máximo (muito curto) para ser reparado. A verificação e teste destes procedimentos eram realizados periodicamente pelo técnico de informática existente em cada entreposto, em particular aos terminais de rádio frequência, impressoras e equipamentos activos da rede.

6.10.1 OS UTILITIES

O primeiro nível de contingência corresponde a um conjunto de equipamentos não nucleares do projecto, mas de fundamental importância na solução final.

A. UPS (UNINTERRUPTED POWER SUPPLY)

Os equipamentos considerados críticos, ou seja, que parando poderiam comprometer o bom funcionamento do sistema informático e, consequentemente as operações do entreposto, foram

ligado a uma UPS. Nesse lote incluem-se entre outros, *switches* de Ethernet, servidores, routers e bastidores de comunicações. A carga das UPS foi dimensionada com base nos equipamentos existentes, com uma folga para a adição de novos. As UPS utilizadas foram ligadas a um gerador interno, para garantir a continuidade do serviço de abastecimento de energia em caso de falha.

B. INSTALAÇÕES

As instalações onde residiram os sistemas Unix nos entrepostos foram, dentro do possível, preparadas tendo em conta os seguintes requisitos:

- Existência de paredes corta-fogo a proteger os sistemas envolvidos.
- Existência de chão falso para proteger a cablagem.
- Utilização de calhas metálicas igualmente para a protecção da cablagem.
- Corredor técnico para uma intervenção dos técnicos de *hardware*, em caso de avaria.
- Não existência de janelas nas salas de sistemas.
- Acesso controlado às salas dos sistemas e, mais particularmente, às suas consolas.
- Não existência de quadros eléctricos, UPS ou outros equipamentos de potência nas salas dos sistemas.
- Instalação de equipamentos de ar condicionado para manutenção do ambiente à temperatura recomendada.

C. COFRE

Para garantir o *backup* diário dos dados foi criado um procedimento de transferência de *tapes* para um cofre *offsite*. Era um cofre comum com as características principais desejáveis: blindado, à prova de fogo, com chave de segurança e com espaço suficiente para assegurar o armazenamento das *tapes*, de acordo com a rotatividade definida.

6.10.2 A REDE DE RÁDIO FREQUÊNCIA

A rede de rádio frequência constitua um componente essencial para as operações do entreposto. Com efeito, todos os terminais portáteis utilizavam este meio físico para comunicar com o sistema central. O desenho da solução final da rede de rádio frequência, ilustrada na Figura 6-15, teve em conta as seguintes preocupações:

- Ter sempre duas antenas a cobrir a mesma área, garantindo dois acessos distintos para a mesma zona.
- Cada ponto de acesso, e consequente antena, estavam ligados a *switches* de Ethernet distintos.
- A alimentação eléctrica era assegurada por fases diferentes, ligadas a UPS igualmente distintas.

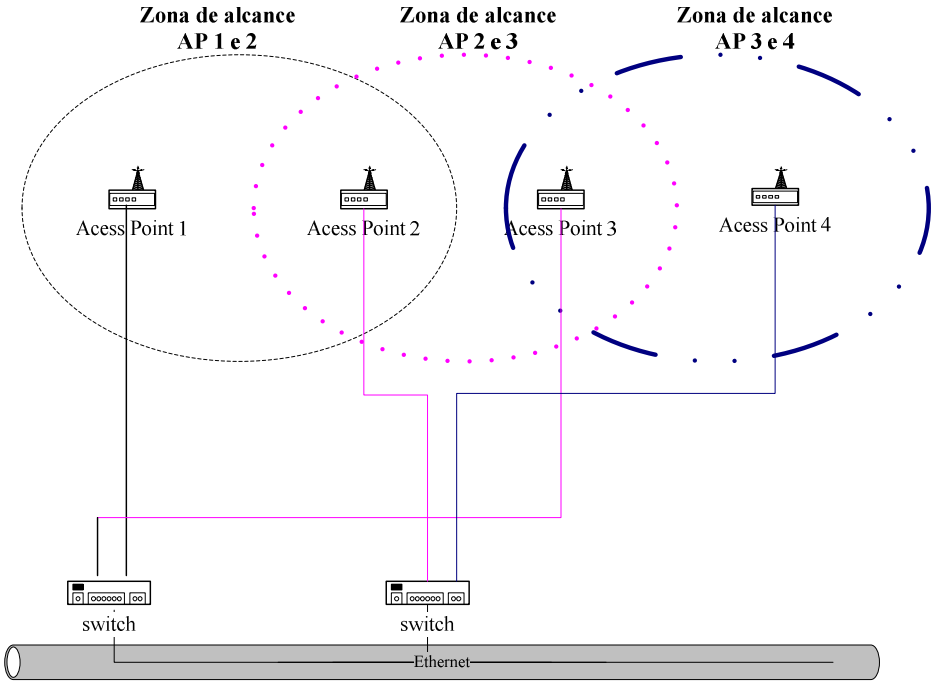


Figura 6-15 – Esquema geral da rede de rádio frequência.

Conforme se pode constatar pela figura, uma determinada zona do entreposto foi abrangida por dois AP. Se o AP 1 avariasse, a área por ele abrangido encontrava-se coberta pelo AP 2. Este procedimento foi utilizado em toda a área do entreposto, garantindo assim total cobertura redundante pela rede de rádio frequência. O estado das antenas era monitorizado por uma aplicação específica disponibilizado pelo fabricante que, de uma forma gráfica, identificava a falta de conectividade aos APs.

Relativamente aos relógios de ponto, o operador poderia utilizar qualquer um, o que minimizava o impacto de avaria. Ainda assim, tanto para os relógios de ponto como para os terminais de rádio frequência, foi mantido em cada entreposto um equipamento suplente para substituição em caso de avaria.

Os pontos críticos de falha da rede local são muito semelhantes aos identificados no Capítulo 4. As medidas adoptadas foram igualmente muito idênticas:

- Interfaces de rede Ethernet. Foram instaladas duas placas de rede *Ethernet* em cada servidor Unix.
- *Switch.* Cada sistema foi ligado a dois *switches* de Ethernet distintos.
- Router. Foram usados dois *routers*, cada um ligado aos dois *switches* de Ethernet.
- Ligação RS232 entre os dois sistemas.

6.10.3 AS IMPRESSORAS

Neste projecto o recurso às impressoras era essencial. Com efeito, nas áreas de expedição eram impressas diariamente centenas de etiquetas, tanto para fixar nas *paletes* a expedir para as lojas e para a área de *picking*, como em cada volume da encomenda quando esta era preparada e

colocada nas *paletes*. Parece um paradoxo afirmar que estes componentes eram críticos no projecto, quando anteriormente foi aferido que o objectivo consistia em eliminar o papel. No entanto, no caso concreto das etiquetas, estas permitiam realizar a recepção da mercadoria na loja de uma forma mais eficiente. O recurso às impressoras era de tal forma crítico que, no limite, uma avaria nos serviços de impressão podia implicar a paragem das operações de recepção e expedição, afectando gravemente o normal funcionamento do entreposto. Nesse sentido, foram tomadas algumas medidas:

- Existência de uma impressora térmica suplente, devidamente configurada e operacional.
- Nalgumas impressoras críticas foi instalado um equipamento que permitia que a ligação ao sistema fosse também realizada através da rede de rádio frequência. Desta forma seria possível ter uma impressora ligada, sem qualquer fio, no meio do entreposto, como se de um terminal de rádio frequência se tratasse.

6.10.4 O SERVIDOR UNIX

Os componentes descritos anteriormente eram importantes, mas os servidores Unix foram, sem dúvida os componentes que requereram maior atenção. Para garantir contingência no sistema Unix a solução consistiu na implementação de uma arquitectura de alta disponibilidade usando um cluster HACMP, com dois nós. A identificação e solução dos pontos críticos de falha seguiram a abordagem descrita anteriormente no Capítulo 4. A metodologia seguida nesta fase foi a seguinte:

1. Levantamento dos pontos críticos de falha.
2. Identificação do grau de risco assumido, embora mantendo a total disponibilidade do sistema.
3. Implementação da infra-estrutura necessária.
4. Implementação do ambiente de alta disponibilidade.
5. Testes de aceitação da arquitectura proposta.
6. Documentação.

No Anexo J é apresentado um esquema com a solução final implementada, ilustrando a mitigação dos pontos críticos de falha mais importantes. Relativamente à localização dos servidores Unix, havia três formas principais de eliminar a sala de computadores como um ponto crítico de falha:

- Os dois servidores serem instalados na mesma sala e fisicamente próximos.
- Ambos os servidores estarem na mesma sala mas separados por uma parede corta-fogo.
- Os dois servidores estarem fisicamente distantes, em edifícios separados.

Comparativamente, os três cenários apresentam algumas diferenças entre si:

- O primeiro cenário é o mais fácil de implementar, não obrigando à instalação de novas infra-estruturas. No entanto, não poderá ser considerada uma solução para o caso de um

desastre. No caso de um incêndio com destruição, os dois sistemas seriam destruídos, sem possibilidade de recuperação.

- No segundo cenário torna-se possível, no caso de um desastre, que o nó de *standby* do cluster possa retomar os serviços num curto espaço de tempo. Nesta solução há a necessidade de implementar algumas infra-estruturas, como é o caso da parede corta-fogo.
- O último cenário é o mais realista em termos de recuperação em caso de desastre. Os riscos são mínimos e a arquitectura é totalmente robusta no que diz respeito à sua recuperação. No entanto, obriga à existência de infra-estruturas mais elaboradas, como sejam, duas salas devidamente equipadas com todas as características desejáveis (ar condicionado, chão falso, paredes corta-fogo, entre outras) e uma ligação remota entre os discos.

O cenário adoptado para este projecto foi o segundo, onde os servidores residiam fisicamente na mesma sala, separados por uma parede corta-fogo. O acesso às instalações foi apenas autorizado por intermédio de código de acesso. Relativamente aos discos externos partilhados, a segunda cópia foi ligada fisicamente num subsistema de discos do nó de *standby*, instalado na sala contígua. A ligação entre as duas cópias foi suportada por uma ligação SSA simples.

6.11 SEGURANÇA

Relativamente ao acesso à aplicação, foram considerados três níveis principais:

- **Operadores** do entreposto, com acesso aos menus autorizados da aplicação, formatados para o terminal que estiverem a utilizar.
- **Equipa de desenvolvimento,** com acesso privilegiado à aplicação. Eram os responsáveis por alterações à parametrização, definição de novos utilizadores e desenvolvimento de novas funcionalidades aplicacionais.
- **Suporte aplicacional**, efectuado remotamente, tinha acesso à aplicação para resolver todos os problemas registados pelos operadores do entreposto.

Relativamente ao acesso iterativo aos servidores AIX, foram definidos os seguintes, com diferentes privilégios:

- **Administração**, responsável pelas tarefas de administração e gestão do sistema. O acesso de `"root"` foi limitado ao mínimo, tendo-se criado utilizadores com contas privilegiadas para o efeito.
- **Gestão de *backups***, para verificação dos *logs* de *backup* dos dados e dos processos de *checkout* e *checkin* das *tapes* na aplicação centralizada de *backups* (Capítulo 7).
- **Gestão**, habilitado a resolver os problemas do dia-a-dia no entreposto, como o controlo de filas de impressão, monitorização do estado do sistema, da base de dados e da aplicação, entre outras tarefas de rotina.
- **Operação TI do entreposto**, tem acesso ao controlo das filas de impressão e visualização dos erros ocorridos no sistema.

Devido ao elevado grau de risco deste projecto, bem como a necessidade de definir um plano de contingência e de *disaster recovery*, o projecto NPO utilizou ainda uma solução de gestão centralizada de *backups*, através da aplicação ADSTAR Distributed Storage Manager (ADSM), detalhada no Capítulo 7.

7 SISTEMA DE GESTÃO CENTRALIZADA DE *BACKUPS*

Durante o desenvolvimento do projecto NPO, descrito no Capítulo 6, decorria em paralelo a implementação de um sistema centralizado de gestão de *backups*, utilizando a aplicação Adstar Distributed Storage Manager (ADSM) [75], que acompanhei enquanto consultor técnico para a área de sistemas Unix. Neste capítulo pretende-se enquadrar os principais conceitos associados a uma arquitectura de *backups* centralizados e as suas principais vantagens. Descreve-se ainda a implementação nos servidores do projecto NPO-Entrepostos e a solução de alta disponibilidade adoptada para a realização de *backups*.

7.1 DESCRIÇÃO DO PROJECTO

O crescimento exponencial do número de utilizadores e da dimensão e heterogeneidade das redes acarretou necessariamente novos níveis de complexidade na gestão e salvaguarda dos conteúdos produzidos. As novas exigências traduziram-se essencialmente na necessidade de passar de um sistema individual de *backups*, efectuado em cada máquina, para um outro centralizado, onde os dados são guardados num repositório único e catalogados para a eventual necessidade da sua reposição. Este novo paradigma deu origem à implementação de soluções centralizadas de *backups*, tirando partido de *hardware* específico baseado em sistemas robotizados, bem como do uso de aplicações de gestão e controlo centralizado de toda a actividade de salvaguarda dos dados.

A rede gerida à altura pela SRD apresentava já um nível de heterogeneidade elevado, envolvendo servidores de vários sistemas operativos, entre os quais AIX, Sun-Solaris, Novell e clientes Windows. Cada servidor dispunha de uma estratégia própria para a realização de *backups*, dependendo da sua utilização. Do ponto de vista de administração, este cenário trazia várias desvantagens:

- Acesso físico aos equipamentos no centro de informática para inserção do dispositivo de *backup* (normalmente uma *tape*).
- Vários esquemas de *backup*. Cada servidor tinha necessidades diferentes, dependendo das aplicações envolvidas e esse facto levada a uma sobrecarga dos técnicos responsáveis pelas operações de *backup*, nomeadamente em períodos nocturnos.
- Visualização de vários ficheiros de *log* individuais relativos às actividades de *backup* de cada sistema. Esta operação aumentava a probabilidade de ocorrência de erros humanos na análise dos ficheiros.
- Dependência das unidades de *backup* de cada sistema. No caso de uma avaria na unidade de *backup* de um sistema, a salvaguarda dos dados não seria efectuada até à sua reparação.

O projecto em curso visava colmatar essas desvantagens através da implementação de uma arquitectura centralizada, assegurando a integração das várias plataformas de sistemas operativos existentes.

7.1.1 OBJECTIVOS

Assim, este projecto era estrutural para a SRD (e posteriormente, para a Novis) na medida em que abrangia a maioria dos servidores geridos e alteraria também alguns procedimentos de gestão. Concretamente, este projecto assentava em dois objectivos base:

- Optimizar os processos de gestão de *backups*, com vista à rápida recuperação dos dados.
- Optimizar os recursos humanos envolvidos nas operações de gestão de *backups*, nomeadamente centralizando o controlo e monitorização das tarefas de *backup* e dos ficheiros de *log*.

Mais especificamente, do ponto de vista da solução técnica adoptada, o projecto visava a execução das seguintes tarefas:

- Instalar e configurar o servidor da aplicação ADSM para o sistema operativo AIX.
- Instalar e configurar um servidor ADSM para o sistema operativo HP-UX, bem como os clientes disponíveis para esta plataforma.
- Instalar e configurar a componente cliente nos vários servidores, nomeadamente Windows, AIX e HP-UX
- Definir políticas de *backup* para cada cliente, de acordo com o grau de criticidade dos dados.
- Realizar testes de recuperação dos dados relativos aos clientes configurados.

A realização quase em simultâneo do projecto NPO permitiu igualmente incluir na lista de objectivos a integração da solução centralizada de *backups* com os planos de contingência definidos nos clusters de alta disponibilidade em HACMP. Com efeito, a unidade de *backup* deixaria de ser local e passaria a ser partilhada pelos nós do cluster, devendo estar disponível no nó que em cada momento assegurava os grupos de recurso. A solução de alta disponibilidade, integrada com a arquitectura adoptada para o ADSM, foi implementada nos clusters HACMP dos três entrepostos.

7.1.2 ENVOLVIMENTO

Este projecto decorreu em simultâneo com o projecto NPO (Capítulo 6) do qual eu era gestor da componente tecnológica. No entanto, integrei também o comité de avaliação técnica da escolha do *hardware* a utilizar, nomeadamente a unidade robotizada, e acompanhei as fases de instalação, configuração e testes à solução adoptada. Este projecto contou com a consultoria externa da empresa "Informática El Corte-Inglês, S.A." e da IBM.

7.2 A ARQUITECTURA DO ADSM-TIVOLI STORAGE MANAGER

A aplicação ADSM, posteriormente designada ADSM-Tivoli Storage Manager (TSM) [76] é uma aplicação cliente-servidor para efectuar a salvaguarda de dados em ambientes multi-plataforma. Para realizar *backups online* de aplicações (por exemplo, bases de dados), o ADSM disponibiliza ainda bibliotecas de programação (API) para o efeito. Por exemplo, para realizar um

backup online de uma base de dados Informix, é disponibilizada a aplicação "OnBar", que efectua a ligação ao servidor ADSM e executa o *backup* das tabelas pretendidas.

Os dados que se pretendam salvaguardar são enviados dos clientes para o servidor ADSM através da rede. O servidor encarrega-se de gerir os dados que recebe, actualizando a sua base de dados e enviando, sempre que necessário, os ficheiros para os suportes magnéticos disponíveis, neste caso do tipo *tape*. A Figura 7-1 pretende ilustrar a arquitectura geral do ADSM.

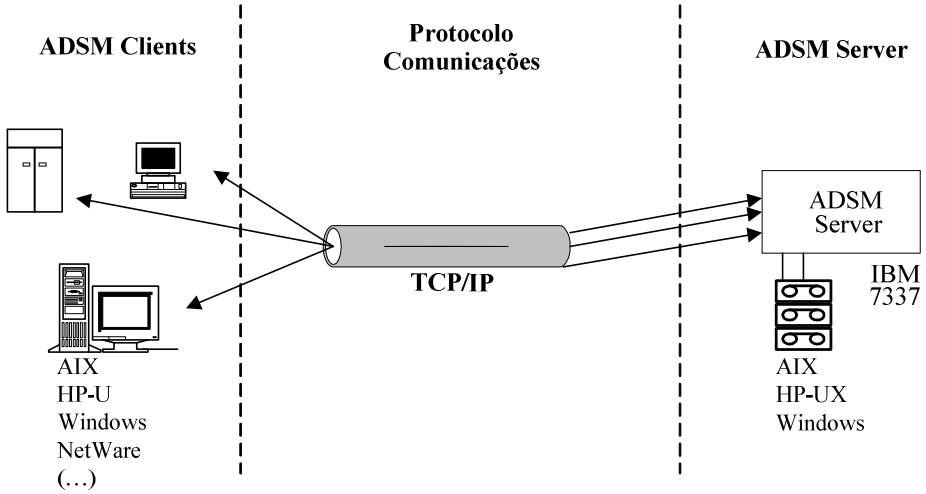


Figura 7-1 – Arquitectura da aplicação ADSM.

Uma plataforma de gestão centralizada de cópias de segurança (*backups*) apresenta os seguintes conceitos importantes relacionados com a gestão de unidades de armazenamento, e que foram incorporados no projecto:

- Os *backups* podem ser incrementais ou selectivos. Os primeiros correspondem ao armazenamento de ficheiros novos ou que foram alterados após o último *backup* total. No caso dos *backups* selectivos, o utilizador pode escolher os ficheiros (directórios ou *filesystems*) que pretende guardar.
- Cada utilizador pode recuperar um ficheiro de um *backup*, solicitando a operação ao servidor de ADSM.
- Cada versão de um ficheiro salvaguardado está sujeito a um período de retenção. Após esse período, o *backup* expira não sendo possível recuperá-lo posteriormente.
- É possível estabelecer um conjunto de regras relativas à forma como o ADSM trata os dados que recebe para salvaguardar. As políticas de *backups* da empresa são centralizadas no servidor de ADSM em estruturas lógicas, denominadas *policy domain*, que correspondem a uma colecção lógica de clientes.
- Dentro de cada *policy domain* são definidas políticas, denominadas de *policy sets.*
- Cada *policy set* é constituído por várias classes, *management classes*, que reflectem a forma como os *backups* dos clientes serão realizados. Desta forma é possível que os dados de tipos diferentes sejam tratados pelo ADSM de forma igualmente distinta.

- Uma *storage pool* consiste num conjunto de *tapes* definida na *mangement class* que irá conter os dados recebidos pelo ADSM dos diversos clientes para serem salvaguardados em *tape*.
- As *tapes* têm obrigatoriamente de estar inicializadas com uma etiqueta (*label*) que é lida pela unidade de leitura (código de barras) da unidade robotizada. Após esta leitura a *tape* é logicamente associada pelo servidor de ADSM aos dados que guardar.
- Uma solução de *disaster recovery* inclui um *backup* da informação da *storage pool*, denominada de *copy pool*. A *tape* com o *backup* desta *pool* deverá ser enviada para um cofre (backup *offsite*), *num* local fisicamente distante do sistema.
- Sempre que se retira uma *tape* da unidade de *backup* para o cofre (*backup offsite*) é necessário enviar essa informação ao servidor de ADSM. A este processo dá-se o nome de *checkout* da *tape*.
- O processo de *checkout* retira lógica e fisicamente a *tape* da unidade de *backup*. O *checkin* ocorre quando se coloca uma nova *tape* na unidade de *backup*, passando o ADSM a reconhecê-la e a disponibilizá-la para operações de *backup*. No processo de *checkin* a nova *tape* deverá ser necessariamente atribuída a uma *pool*.

7.3 COMPONENTES DO ADSM

O ADSM é constituído por três componentes principais [75], ilustrados na Figura 7-2:

- **Clientes** - PC ou servidor, que têm a informação crítica para ser salvaguardada pelo ADSM.
- **Servidor -** responsável por receber os dados dos clientes e armazená-los em *storage pools*. O ADSM utiliza uma base de dados para guardar a localização dos dados na *storage pool*.
- **Administrador -** controla o servidor de ADSM. É o administrador de ADSM que define as políticas de salvaguarda de dados e correspondentes regras que especificam de que forma serão tratados durante uma operação de *backup*.

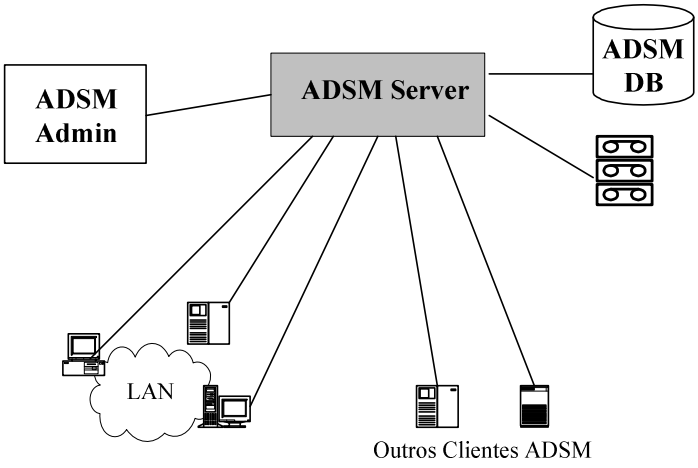


Figura 7-2 – Interligação dos componentes da aplicação ADSM.

Sobre o *hardware* envolvido, foi utilizada uma unidade de armazenamento robotizada IBM 3494 ilustrada na Figura 7-3.

Retiradas de http://www.ibm.com

Figura 7-3 – A unidade robotizada de *backup* IBM 3494.

Esta unidade foi ligada directamente por SCSI ao nó do SP onde estava já instalado o servidor de ADSM. A unidade adquirida tinha seis *drives* IBM 3590-B1A, possibilitando assim o *backup* em simultâneo de seis clientes ADSM. Esta unidade da IBM fornecia ainda algumas características importantes do ponto de vista de administração:

- Implementava mecanismos intrínsecos de alta disponibilidade e tolerância a falhas nos seus componentes.
- Assegurava a conectividade a várias plataformas de sistemas operativos.
- Possuía um crescimento modular.
- Dispunha de uma interface gráfica de administração, que permitia o controlo da actividade do robot e dos processos de *checkin* e *checkout* das *tapes*.

7.4 O ADSM NO PROJECTO NPO

O processo de *backup* utilizando uma solução centralizada não se traduziu apenas na alteração ao modo como as *tapes* eram geridas. Obrigou, acima de tudo, a uma nova forma de encarar a problemática da salvaguarda dos dados, com vista à definição de estratégias eficientes de *backups* e *disaster recovery*.

No caso concreto do projecto NPO procedeu-se à seguinte sequência de tarefas:

- Inventariação de toda a informação existente no sistema.
- Agrupamento, em conjuntos, dessa informação por temas e graus de importância.
- Definição de políticas de *backup* para cada um dos conjuntos, com a parametrização necessária, de acordo com as necessidades do projecto.

Foi instalado um servidor de ADSM em cada sistema Unix existente nos entrepostos. No fundo, o cliente e o servidor de ADSM residem no mesmo sistema, tratando-se, portanto, de uma arquitectura um pouco mais simplificada [69].

Por se tratar de uma aplicação bastante transaccional, onde os dados gerados nas operações do entreposto consistiam essencialmente em documentos, foi assumido que o tempo de vida máximo dos *backups* seria de 30 dias.

Cada sistema Unix dispunha de uma unidade robotizada de *backup* IBM 7337, com duas *drives* de leitura e escrita. Esta unidade suportava no máximo 15 *tapes* DLT IV, com capacidade até 70 GB cada. A ligação ao sistema Unix era assegurada através de uma ligação SCSI diferencial.

Em termos de *pools*, foram definidas duas:

- *storage pool* com 5 *tapes* DLT IV para os *backups* diários.
- *copy pool* que, diariamente, enviava a informação da *storage pool* para uma *tape* que, posteriormente, era colocada *offsite*, num cofre.

No final de cada mês era realizada uma operação de reorganização da base de dados do ADSM, denominada de *reclaim storage*. Esta operação percorria os ficheiros inventariados nas *tapes* utilizadas nesse mês, procedendo ao seu envio para uma outra, correspondendo ao *backup* mensal. As cinco *tapes* utilizadas na *storage pool* salvaguardavam:

- O primeiro *backup* total do sistema.
- Os *backups* incrementais diários.
- Os ficheiros de transacções da base de dados (*logical logs*).
- Outros *backups* temporários e pontuais que se pretendiam guardar no servidor ADSM.

Considerando a necessidade de implementar uma solução de *disaster recovery*, foi igualmente realizado o *backup* à base de dados do próprio servidor ADSM, bem como da sua configuração logo após o *backup* diário. Este *backup* era enviado para uma unidade de *tape* independente da anterior, através de comandos de *backup* do Unix.

Neste projecto foram utilizadas apenas dez posições para as *tapes* DLT IV, das quinze disponíveis. Dessas dez, cinco foram usadas para a *storage pool*, três para a *copy pool* e as restantes duas posições foram reservadas para a necessidade de as alocar a alguma das *pools* anteriores.

A distribuição das *tapes* no robot da unidade IBM 7337 está ilustrada na Figura 7-4.

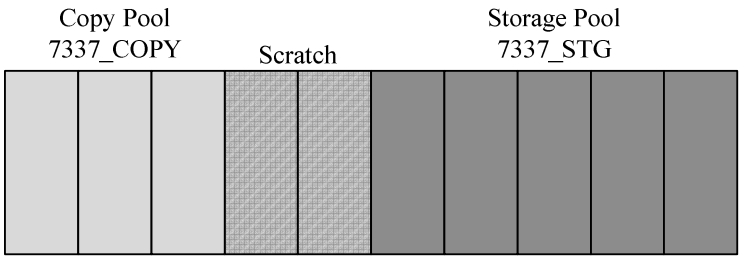


Figura 7-4 – Disposição das *tapes* na unidade de *backup* IBM 7337.

Foi definida a seguinte estratégia de *backups* para o projecto:

- Durante o dia:
  - *Backup* dos ficheiros de *logging* da Base de dados.
  - Outros *backups* temporários.
- À noite, escalonado na `cron` do sistema:
  - *Backup online* da base de dados.
  - *Backup* incremental de todo o sistema para a *storage pool*.
  - *Backup* da *copy pool,* automaticamente pelo ADSM para a *tape* do *backup offsite.*
  - *Backup* da base de dados do ADSM e da sua configuração para uma segunda *tape* a colocar posteriormente no cofre *(offsite).*
- No início do dia seguinte, o operador realizava as seguintes operações:
  - *Checkout* da tape existente na *copy pool* e posterior arquivo *offsite,* juntamente com a que contém a base de dados do ADSM e a sua configuração.
  - *Checkin* de uma nova *tape* a utilizar no próximo *backup* diário na *copy pool.*
- No inicio de cada mês, após o *backup* diário:
  - *Reclaim storage* da informação existente na *copy pool* para uma nova *tape* a colocar *offsite.* Era igualmente efectuado o processo de expiração de versões de alguns ficheiros.
  - Reutilização das 30 *tapes,* colocando-as em modo de leitura/escrita.

7.5 O ADSM E A SOLUÇÃO DE ALTA DISPONIBILIDADE

Um dos objectivos dos *backups* é ser possível recuperar rapidamente as aplicações do negócio no caso de um desastre que destrua os dados existentes nos discos. Dessa forma, pretendeu-se que o nó de *standby* do cluster HACMP pudesse também estar ligado à unidade de *backup*. Primeiro, para também realizar *backups* quando assumisse o papel de nó principal. Depois, pensando numa solução de *disaster* recovery, para que se pudesse iniciar a recuperação dos dados tão rápido quando possível após um desastre.

Há várias abordagens que podem ser consideradas. No projecto NPO assumiram-se as seguintes opções, cuja arquitectura genérica se ilustra na Figura 7-5 [69]:

- A unidade IBM 7337 foi ligada fisicamente aos dois sistemas Unix, através de um cabo SCSI diferencial, com uma configuração em "Y".
- O ADSM foi instalado nos dois sistemas. O de *standby* do HACMP assumiu o papel de servidor de ADSM (e cliente também), enquanto o sistema principal foi apenas cliente. Diariamente, o cliente (nó principal) efectuava o procedimento de *backup* para o sistema servidor de ADSM (nó de *standby*).

- Em caso de um desastre no nó primário, a unidade IBM 7337 continuaria disponível a partir do nó de *standby* bem como o servidor de ADSM. Em termos de plano de contingência, o nó de *standby* e a unidade IBM 7337 ficaram, fisicamente, numa sala protegida isolada com parede corta-fogo.

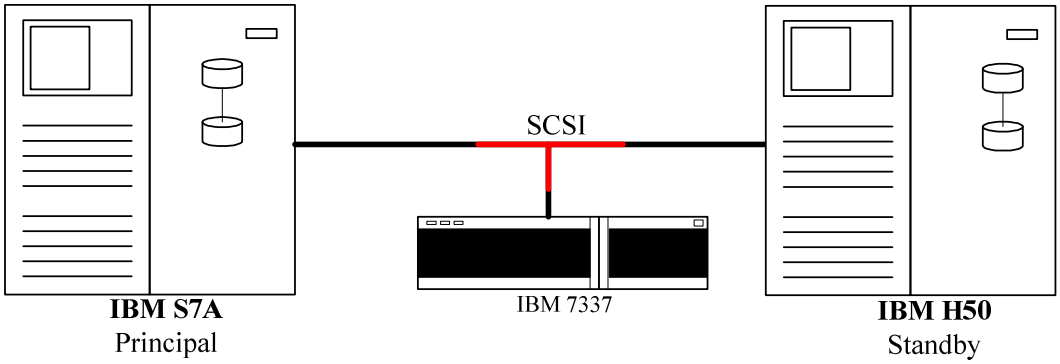


Figura 7-5 – Arquitectura de contingência das unidades de *backup* no projecto NPO.

As soluções de centralização de *backups* podem constituir um valor acrescentado para projectos de média ou grande dimensão. Em projectos críticos e de grande impacto no negócio, esse facto torna-se ainda mais importante. Dessa forma, justifica-se o investimento numa ferramenta de gestão de *backups* que:

- Seja independente da plataforma.
- Automatize diversas operações de inventariação e registo do conteúdo dos suportes magnéticos.
- Implemente características conducentes à definição de soluções de *disaster recovery* fiáveis, sendo possível recuperar um sistema após uma catástrofe.
- Permita recuperar ficheiros isolados, devido a um erro humano, de forma fácil e rápida.

No entanto, soluções como o ADSM envolvem custos significativos a vários níveis, nomeadamente:

- Aquisição do ADSM e respectivas licenças.
- Consultoria na definição de políticas de *backup* globais e desenho de soluções de *disaster recovery*.
- Aquisição de unidades de *tapes* rápidas e com capacidade suficiente para corresponder às necessidades do projecto em causa.
- Formação específica na administração, gestão e operação do ADSM.

Estes custos terão de ser bem equacionados, atendendo ao risco que se pode correr e à necessidade de recuperar rapidamente os dados. A experiência adquirida permite afirmar que se trata de uma arquitectura que traz valor acrescentado à gestão da rede, conferindo maior fiabilidade aos utilizadores e prevenção redobrada no caso de uma catástrofe. Quando se implementa uma solução de centralização de *backups* é importante reconhecer que:

- A maneira como a empresa e as pessoas envolvidas no projecto passam a olhar a gestão de *backups* muda completamente a vários níveis, nomeadamente na forma de gerir e administrar as rotinas diárias de *backup*.
- Este tipo de soluções acrescenta vários conceitos novos e paradigmas de gestão de *backups* igualmente inovadores. Dessa forma, há necessidade de formar técnicos em ADSM, ao nível da operação, administração e gestão.
- A automatização de tarefas é importante e pode contribuir para a diminuição de erros nas tarefas de operação do ADSM.

A construção de centros de informática que assegurem a recuperação de dados em caso de catástrofe envolve custos avultados. Dependendo da criticidade do negócio, as empresas normalmente têm acordos com centros de recuperação de dados, em complemento à estratégia de *backups* que implementam internamente. Neste tipo de centros encontram-se os de Business Continuous Recovery Service da IBM [26]. Em Portugal há dois centros dedicados a este fim e o objectivo é assegurar a duplicação *offsite* das transacções diárias das empresas, bem como a operacionalidade do negócio no mínimo espaço de tempo, em caso de um desastre com destruição dos servidores e até das instalações.

8 CONCLUSÕES

Neste relatório foram apresentadas três áreas de actuação onde adquiri experiência consolidada no período compreendido entre 1988 e 2001: a gestão e administração de redes, servidores e serviços; a implementação de soluções de alta disponibilidade usando clusters de servidores AIX e Linux; a gestão e coordenação de projectos de tecnologias de informação. Foi apresentada a topologia das redes geridas, evidenciando as principais tecnologias de computadores, de sistemas operativos e de redes. Foram igualmente enquadradas as principais tecnologias e conceitos associados aos projectos descritos no relatório. Foram ainda evidenciadas as principais características distintivas de cada projecto e o seu enquadramento no contexto do grupo Sonae. As conclusões descritas de seguida dividem-se em três tipos principais: pessoais, organizacionais e técnicas.

Ao nível pessoal, importa destacar o enriquecimento obtido durante o período, resultado de vários factores. Em primeiro lugar, devido às facilidades que foram sempre concedidas para realizar uma gestão de carreira ajustada às motivações pessoais e profissionais em cada momento. De seguida, pela possibilidade de ter integrado equipas multidisciplinares, fortemente motivadas para contribuir para objectivos globais (da empresa ou do departamento). Por fim, a possibilidade de desenvolver a actividade profissional usando a criatividade, contribuiu igualmente para a valorização pessoal.

Ao nível organizacional, destaco o facto de ter integrado empresas cuja estratégia consistia em investir na inovação e no desenvolvimento tecnológico, permitindo uma actualização constante de conhecimentos.

Por fim, ao nível técnico, o caminho percorrido foi muito positivo. Por um lado, foi possível contactar de perto e, nalguns casos em primeira mão, com uma miríade de tecnologias de informação e comunicação, nomeadamente sistemas operativos, arquitecturas de computadores e aplicações. Por outro, os constantes desafios propostos ao longo do percurso revelaram-se sempre muito motivadores para apostar tecnicamente na área, destacando-se os vários projectos que tive oportunidade de coordenar.

No último trimestre de 2000 dei início à preparação de uma alteração significativa na minha carreira profissional, que se viria a verificar em Março de 2001. Esta mudança consistiu no início de uma actividade docente no ensino superior politécnico público e deveu-se essencialmente à forte motivação pessoal para encetar uma carreira académica, de cariz prático e laboratorial, como é o caso do ensino superior politécnico.

8.1 PONTO DE SITUAÇÃO ACTUAL

Em termos gerais, a realidade actual do contexto empresarial, das tecnologias e dos projectos apresentados neste relatório, é naturalmente diferente, fruto essencialmente do desenvolvimento tecnológico e organizacional observado desde 2001. No entanto, o trabalho descrito, os conhecimentos aplicados e os manuais e procedimentos de boas práticas que foram desenvolvidos,

mantém-se constantes e serviram de guia para a implementação de vários projectos posteriores a 2001.

Relativamente ao contexto empresarial, realçam-se as alterações que se descrevem de seguida. A estrutura organizativa do grupo Sonae sofreu várias alterações, estando agora dividida da forma como se ilustra na Figura 8-1 [4].

Figura 8-1 – Áreas de negócio actuais do grupo Sonae [8].

Conforme se pode constatar, o diagrama organizacional do grupo é hoje bem diferente daquele que existia no final da década de 90. As áreas de negócio definidas actualmente são quatro: negócios "core", parcerias "core", negócios relacionados e investimentos activos. Os negócios "core" correspondem ao retalho alimentar (Sonae MC) e às insígnias com posições de referência nos respectivos segmentos de mercado (SonaeSR). Conta ainda com duas parcerias nucleares: gestão de centros comerciais (SonaeSierra) e telecomunicações e serviços de gestão de serviços e tecnologias de comunicações (SonaeCom) [77]. As restantes duas áreas de negócio incluem empresas de gestão de investimentos e de imobiliário do grupo.

A indústria é ainda um sector de actividade valorizado pelo grupo, conforme se constata pela internacionalização que decorreu nos últimos anos, com a implantação de fábricas por vários países, como Brasil, Espanha e Canadá [78]. A Novis passou a ser apenas uma marca, tendo sido criada uma nova empresa, a Mainroad (inserida na área de negócios da SonaeCom), responsável actualmente pela gestão da infra-estrutura de redes e serviços do grupo. Devido à política de expansão encetada pela Novis, a Mainroad tem actualmente um leque de clientes mais abrangente, não se limitando às empresas do grupo Sonae. Por fim, o principal cliente da Novis (e ainda da Mairoad), a Modelo Continente Hipermercados S.A., criou uma equipa de gestão e administração da sua infra-estrutura de redes e serviços, tendo absorvido parte dos técnicos daquela.

Sobre a rede da Sonae e a sua configuração actual, as mudanças são visíveis essencialmente ao nível tecnológico. Genericamente, as mudanças estão em linha com os avanços tecnológicos verificados nos últimos anos por parte dos construtores de tecnologia. Relativamente ao SP, foi

[8] Retirado de http://www.sonae.pt/pt/sonae/areas-de-negocio/

descontinuado pela Sonae em meados de 2007, sendo actualmente utilizadas arquitecturas mais flexíveis, baseadas no particionamento de servidores em vários sistemas operativos.

Os servidores Unix utilizados no projecto NPO-Entrepostos sofreram actualizações de configuração, acompanhando os desenvolvimentos tecnológicos operados pela IBM. Também a unidade robotizada de *backups* IBM 3494 sofreu uma expansão considerável para acoplar os vários servidores que foram integrados na rede. A unidade robotizada de backup tem actualmente 12 drives (143 MB/s cada). As *tapes* usadas actualmente têm a capacidade nativa de 640 GB, embora possam alojar cerca de 1.8 TB com uma compressão de 3:1. Paralelamente, foi igualmente instalado um sistema VTL (Virtual Tape System) que, através da virtualização permite aumentar significativamente a capacidade disponível.

Relativamente às aplicações, foram gradualmente substituídas por novas versões, com paradigmas mais actuais. No caso do EDI, é actualmente utilizado o Biztalk para a integração dos dados com os fornecedores. Inclui igualmente uma plataforma de processamento de faxes, para onde os documentos (encomendas, facturas) são enviados, após terem sido preparados e formatados [54, 79].

Relativamente aos projectos descritos neste relatório, à parte das actualizações realizadas no *hardware* e *software* (sistema operativo e aplicacional), as soluções adoptadas na altura, no domínio da alta disponibilidade e da gestão centralizada de *backups*, mantém-se em funcionamento. Os projectos TEDI e NPO-Entrepostos continuam a ser estruturais na cadeia de abastecimento das lojas de retalho da Sonae. O seu grau de criticidade continua elevado e a solução de HACMP instalada mantém-se operacional. Actualmente, outras infra-estruturas aplicacionais assentes em AIX beneficiam deste tipo de soluções de alta disponibilidade usando HACMP, tendo a sua implementação usufruído do trabalho de investigação aplicada, levada a cabo nos projectos descritos neste relatório.

8.2 INTEGRAÇÃO COM A ACTIVIDADE DOCENTE

Desde 2001 tenho sido responsável por várias unidades curriculares nas áreas temáticas de redes, sistemas distribuídos e administração de sistemas, no Instituto Superior de Engenharia de Coimbra (ISEC) e na Escola Superior de Tecnologia e Gestão (ESTG), unidades orgânicas dos Institutos Politécnicos de Coimbra e Leiria, respectivamente. Actualmente sou responsável pelas unidades curriculares de Tópicos Avançados de Redes (TAR) e Gestão e Administração de Redes e Serviços (GARS), da licenciatura em Engenharia Informática da ESTG do Instituto Politécnico de Leiria (IPL), onde lecciono as componentes teórica e prática. Durante este período tenho procurado integrar no contexto académico a experiência empresarial adquirida, bem como desencadear várias acções de interligação entre as duas realidades, conforme se descreve de seguida.

Relativamente à actividade lectiva, destaco a integração de mais alguns tópicos relacionados com a implementação de soluções redundantes e tolerantes a falhas no programa das unidades curriculares de que tenho sido responsável.

No âmbito da integração dos alunos no contexto empresarial, organizei várias visitas de estudo a centros de informática de organizações de referência. Estas visitas foram de cariz técnico e guiadas, tendo permitido aos alunos a consolidação dos conhecimentos adquiridos ao longo do curso sobre estas temáticas. Dessas visitas destaco as seguintes:

- Centros de Business Continuous Recovery Service (BCRS) da IBM, de Lisboa e Porto (quatro visitas).
- Centro de Informática da Mainroad (duas visitas).
- Instituto Português de Comunicações.
- Instalações da Portugal Telecom em Sesimbra.

Na orientação de projectos da Licenciatura em Informática e Comunicações da ESTG-Leiria, orientei vários projectos sobre esta temática destacando dois:

- *"Clusters de alta disponibilidade – uma abordagem open source"*; 2005
- *"Análise de soluções open source para administração de redes", 2007*

Em 2008 propus e orientei um estágio de final de curso realizado na Direcção de Sistemas de Informação da Modelo-Continente Hipermercados, designado "GesFO – ferramenta de gestão centralizada de *front office*". O trabalho de estágio consistiu no desenvolvimento de uma aplicação de gestão centralizada de *front office*, nomeadamente dos pontos de venda (PoS) das frentes de caixa dos hipermercados Modelo e Continente.

Ainda no domínio da gestão de redes e no estudo de tecnologias emergentes, dinamizei a criação de dois grupos de trabalho específicos:

- IPv6@ESTG-Leiria, acessível em http://www.ipv6.estg.ipleiria.pt. Este projecto iniciou-se em 2005 com a criação de uma rede piloto de IPv6 [80]. Posteriormente foram desenvolvidos outros projectos relacionados com o tema, todos com o mesmo denominador comum: o IPv6. Foram igualmente configuradas duas ligações em IPv6 à FCCN: uma nativa e outra recorrendo a um túnel (IPv4-IPv6) configurado para o efeito.
- Movimento Open-Source da ESTG-Leiria (MOSEL), acessível a partir do endereço http://mosel.estg.ipleiria.pt. Este projecto foi motivado pela necessidade de divulgar as aplicações *open source* e a sua aplicabilidade no domínio das unidades curriculares leccionadas na ESTG-Leiria.

Em 2004 dinamizei a criação de uma academia Cisco no ISEC, onde era docente. Obtive a certificação de formação CCNA e colaborei activamente nas acções de construção da academia Cisco. Além da instalação dos componentes tecnológicos, participei ainda na integração do plano de estudos do CCNA na Licenciatura em Engenharia Informática e de Sistemas do ISEC [80].

Durante este período destaco ainda os seguintes projectos realizados em consórcio com a indústria:

- Fevereiro a Dezembro de 2009: através de um concurso de ideias promovido pela PT-Inovação, integrei a equipa de desenvolvimento (IT-Leiria e ESTG-Leiria) de uma plataforma multi-funcional de gestão de frotas.

- Setembro de 2002 a Junho de 2003: participei no projecto MIDWTELCO, na empresa CriticalSoftware. Este projecto consistia na implementação de uma plataforma de *middleware* para a integração e mediação de aplicações em sistemas de telecomunicações.

Como coordenador das relações internacionais do Departamento de Engenharia Informática e Sistemas do ISEC, coordenei os projectos de estágio de vários alunos da Licenciatura, realizados em países da Europa, destacando-se os que decorreram na Philips (Hamburgo, Alemanha) e na Koninklijk Nederlands Meteorologisch Instituut-KNMI (Utrecht, Holanda).

Na componente de investigação, obtive em Outubro de 2001 o grau de Mestre em Informática, no ramo de sistemas e redes. A tese foi realizada na área de administração de sistemas Unix, designadamente no desenvolvimento de uma aplicação centralizada de gestão de grupos heterogéneos de sistemas Unix, cujos grupos partilham características comuns. A plataforma é baseada numa interface Web e suporta vários sistemas operativos Unix [81, 82].

Actualmente sou investigador no centro de investigação CRACS (Center for Research in Advanced Computing Systems), laboratório da Universidade do Porto e associado do INESC-Porto. Nessa qualidade desenvolvo os trabalhos de investigação conducentes à realização da tese de doutoramento. A actividade de investigação tem-se centrado na área da segurança em sistemas e redes, mais concretamente no estudo de algoritmos imuno-inspirados e na sua implementação em sistemas de detecção de intrusões em redes.

BIBLIOGRAFIA

[1] A. Frisch, *Essential system administration, 3rd*: O'Reilly; ISBN 978-0596003432, 2002.

[2] M. Burgess, *Principles of network and system administration*, 2nd ed: Wiley; ISBN: 978-0470868072, 2004.

[3] M. Burgess, "On the theory of system administration," *Science of Computer Programming - Elsevier*, vol. 49, pp. 1-46, 2003.

[4] Sonae, "Sonae - Apresentação de resultados 2009," http://www.sonae.pt/pt/investidores/dados-financeiros/apresentacao-de-resultados/ (acedido em Maio 2010).

[5] IBM, "IBM Archives: IBM System/36," http://www-03.ibm.com/ibm/history/exhibits/rochester/rochester_4018.html (acedido em Março de 2010).

[6] R. L. Hoffman and F. G. Soltis, "The IBM SYSTEM/38: Hardware organization of the SYSTEM/38," *Seiwiorek et al.[26]*, pp. 544–546, 1980.

[7] G. G. Henry, "Introduction to IBM System/38 architecture," *IBM System/38 Technical Developments (1978). IBM Corporation, Atlanta, GA*, 1978.

[8] F. G. Soltis, *Fortress Rochester: the Inside Story of the IBM ISeries*: 29th Street Press; ISBN: 978-1583040836, 2001.

[9] L. J. Kenah, R. Goldenberg, and S. F. Bate, *VAX/VMS internals and data structures*: Digital Press Newton, MA, USA; ISBN: 978-1555580599, 1988.

[10] H. M. Levy and P. H. Lipman, "Virtual Memory Management in the VAX/VMS Operating System," *Computer*, vol. 15, pp. 35-41, 1982.

[11] J. Martin and J. Leben, *DECnet Phase V: an OSI implementation*: Butterworth-Heinemann Newton, MA, USA; ISBN 1555580769, 1991.

[12] S. W., Y. Wang, and J. B. Lindberg, "HP-UX: Implementation of UNIX on the HP 9000 Series 500 Computer Systems," *Hewlett-Packard Journal*, vol. 35, 1984.

[13] J. Martin and K. K. Chapman, *SNA: IBM's networking solution*: Prentice-Hall, Inc. Upper Saddle River, NJ, USA; ISBN: 0-13-815143-1 1987.

[14] C. Janacek and D. Snow, *DB2: Universal Database Certification Guide*: Prentice Hall PTR Upper Saddle River, NJ, USA; ISBN:0130796611, 1997.

[15] D. Quintero, C. Courty, C. Holban, and W. C. W. III, *RS/6000 SP and Clustered IBM eServer pSeries Systems Handbook*: IBM-SG24-5596-02; ISBN-0738422991; IBM Redbooks, 2001.

[16] Y. Kosuge and J. Ellsworth, *SP Perspectives: A New View of Your SP System*: IBM-SG24-5180-00; ISBN-0738412465; IBM Redbooks, 1999.

[17] A. Farazdel, G. R. Archondo-Callao, E. Hocks, T. Sakachi, and F. Vagnini, *Understanding and Using the SP Switch*: IBM-SG24-5161-00; ISBN-0738412864; IBM Redbooks, 1999.

[18] M. Barrios, P. Aguiar, and S. Rabbi, *PSSP Version 3 Survival Guide*: IBM-SG24-5344-00, ISBN: 0738415065, IBM Redbooks, 2000.

[19] M. Barrios, *GPFS: A Parallel File System*: IBM-SG24-5165-00; ISBN-0738403083; IBM Redbooks, 1998.

[20] M. Antunes, "GPFS - Teste Laboratorial e Relatório de Conclusão," Relatório técnico - Novis Departamento de Planeamento e Desenvolvimento de Tecnologias de Informação 2000.

[21] E. Marcus and H. Stern, *Blueprints for high availability*, 2nd ed: Wiley Publishing Inc.; ISBN-978-0-471-43026-1, 2003.

[22] C. Oggerino, *High Availability Network Fundamentals*: Cisco Press; ISBN: 978-1587130175, 2001.

[23] K. Schmidt, *High Availability and Disaster Recovery: Concepts, Design, Implementation*: Springer Berlin Heidelberg, ISBN: 978-3642063794, 2009.

[24] P. S. Weygant and J. Pisoni, *Clusters for High Availability: A Primer of HP Solutions*, 2nd ed: Prentice Hall PTR Upper Saddle River, NJ, USA; ISBN: 0-13-089355-2, 2001.

[25] E. Vargas, "High availability fundamentals," *Sun Blueprints series, Part No.: 806-6857-10*, 2000.

[26] S. Racherla and B. Dufrasne, *IBM System Storage Business Continuity Solutions Overview*: IBM-REDP-4516-00; IBM Redbooks, 2009.

[27] C. Preston, *Backup and Recovery - inexpensive backup solutions for open systems*: O'Reilly; ISBN: 978-0-596-10246-3, 2006.

[28] P. Corcoran, "IBM business continuity and recovery services," *Disaster Recovery Journal*, vol. 16, pp. 23-24, 2003.

[29] W. Green and B. Lancaster, "Carrier-grade: five nines, the myth and the reality," *Pipeline*, vol. 3, 2006.

[30] D. Thiessen, *An HACMP Cookbook*: IBM-SG24-4553-00; ISBN-0738409197; IBM Redbooks, 1995.

[31] S. Horman, "Creating Linux Web Farms (Linux High Availability and Scalability)," *http://verge.net.au/linux/has/ (acedido em Maio 2010)*, 2000.

[32] S. Horman, "Linux Virtual Server Tutorial," *Presented at Linux Simposium, Ottawa, Canada*, 2004.

[33] S. Horman, "Active-Active Servers and Connection Synchronisation for LVS," *Proceedings of Linux Conference - Adelaide - Australia*, 2004.

[34] G. Attebury and B. Ramamurthy, "Router and Firewall Redundancy with OpenBSD and CARP," *IEEE International Conference on Communications (ICC '06)*, vol. 1, pp. 146 - 151 2006.

[35] B. Potter, "Open source firewall alternatives," *Network Security - Elsevier*, vol. 2006, pp. 16-17, 2006.

[36] R. McBride, "Pfsync: Firewall Failover with pfsync and CARP," http://www.countersiege.com/doc/pfsync-carp/ (acedido em Abril 2010).

[37] T. L. Sterling, *Beowulf cluster computing with Linux*: The MIT Press; ISBN: 978-0-262-69292-2, 2001.

[38] S. Bodily, "FLASH10630: HACMP 5.4.1 for Linux is now available," *IBM Technical Documentation*, 2008.

[39] J. R. Shapiro and M. Policht, *Building High Availability Windows Server (TM) 2003 Solutions (Microsoft Windows Server System Series)*: Addison-Wesley Professional; ISBN:0321228782 2004.

[40] J. Kelbley, M. Sterling, and A. Stewart, *Windows Server 2008 Hyper-V: Insiders Guide to Microsoft's Hypervisor*: Sybex; ISBN: 978-0470440964, 2009.

[41] T. Li, B. Cole, P. Morton, and D. Li, "RFC 2281: Cisco Hot Standby Router Protocol (HSRP)," *RFC Editor United States*, 1998.

[42] R. Hinden, "RFC3768: Virtual Router Redundancy Protocol (VRRP)," *RFC Editor United States*, 2004.

[43] P. J. Conlan, *Cisco Network Professional's Advanced Internetworking Guide (CCNP Series)*: Sybex; ISBN: 978-0470383605, 2009.

[44] D. Mills, D. Plonka, and J. Montgomery, "Simple network time protocol (SNTP) version 4 for IPv4, IPv6 and OSI," RFC 2030, October 1996, 2006.

[45] D. L. Mills, "RFC 1305: Network Time Protocol (Version 3)-Specification, Implementation and Analysis," RFC Editor United States, 1992.

[46] D. Plummer, "RFC 826: Ethernet Address Resolution Protocol - Or Converting Network Protocol Addresses to 48 bit Ethernet Address for Transmission on Ethernet Hardware " *RFC Editor United States*, 1982.

[47] F. Ferreira and N. Santos, "Clusters de elevada disponibilidade - uma abordagem open source," Instituto Politécnico de Leiria (ESTG/DEI)- Projecto de final de curso em Engenharia Informática e Comunicações, (orientado por M. Antunes), 2005.

[48] M. Antunes, "Sistemas de elevada disponibilidade - High availability cluster multi-processing," Technical Report: Sonae Redes Dados, S.A. & ISEP-Instituto Politécnico do Porto 1997.

[49] N. Jilovev, *A to Z of EDI and Its Role in E-Commerce*, 2nd ed: 29th Street Press; ISBN: 978-1882419166, 1998.

[50] A. Nevalainen, *The E-Business Dictionary: EDI, Supply Chain, and E-Procurement Terminology*: Rockbend Books Llc; ISBN: 978-0971320321, 2003.

[51] P. D. Obal, *Warehouse & Logistics Software Directory, WMS*, 7th ed: Industrial Data & Information; ISBN: 978-0966934519, 2002.

[52] M. M. Silva, *Integração de Sistemas de Informação*: FCA; ISBN: 978-972-722-391-6, 2003.

[53] G. Stylios, *Principles of Electronic Data Interchange in the Retail Industries*: Prentice Hall; ISBN: 978-0137449057, 1992.

[54] M. Beckner, *Pro EDI in BizTalk Server 2006 R2: Electronic Document Interchange Solutions*, 1st ed: Apress - ISBN: 978-1-59059-935-8, 2007.

[55] O. Lascu, S. Bodily, M.-K. Esser, M. Herrera, P. Pothier, D. Prelec, D. Quintero, K. Raymond, V. Sebesteny, A. Socoliuc, and A. R. Steel, *Implementing High Availability Cluster Multi-Processing (HACMP) Cookbook*: IBM-SG24-6769-00; ISBN-0738494143; IBM Redbooks, 2005.

[56] S. Kannan, A. Demeter, A. Steel, H. Fukuma, and J. Kwecko, *Configuring Highly Available Clusters Using HACMP 4.5*: IBM-SG24-6845-01; ISBN-0738427292; IBM Redbooks, 2002.

[57] O. Conradsen, C. Collin, F. Sabatti, and J. Wang, *Disaster Recovery Using HAGEO and GeoRM*: IBM-SG24-2018-01; ISBN-0738416061; IBM Redbooks, 2000.

[58] IBM, *HAGEO for AIX V2.4 Concepts and Facilities*, vol. Edition number - IBM SC23-1922-04: IBM, 2003.

[59] IBM, *HAGEO for AIX V2.4 Planning and Adminstration Guide*: IBM-SC23-1886-04, 2003.

[60] O. Conradsen, A. Lucenti, M. I. A. Rahman, and V. Sebesteny, *HACMP/ES Customization Examples*: IBM-SG24-4498-01; ISBN: 0738417130; IBM Redbooks, 2000.

[61] E. Chiakpo, *Implementing High Availability on RISC/6000 SP*, vol. IBM-SG24-4742-00; ISBN-0738409553: IBM Redbooks, 1996.

[62] D. Quintero, Z. Borgosz, A. Botura, D. Gilchrist, S. Kister, O. Lascu, and K. So, *RS/6000 SP Cluster: The Path to Universal Clustering*: IBM-SG24-5374-01; ISBN-0738422061; IBM Redbooks, 2001.

[63] A. Robertson, "The evolution of the Linux-HA project," *UKUUG LISA/Winter Conference High-Availability and Reliability*, 2004.

[64] W. Zhang, "Linux virtual server for scalable network services," *Proceedings of Ottawa Linux Symposium*, pp. 1-10, 2000.

[65] W. Zhang, S. Jin, and Q. Wu, "Creating linux virtual servers," *Proceedings of LinuxExpo 1999 Conference*, 1999.

[66] A. Robertson, "Linux-ha heartbeat system design," *Proceedings of the 4th Annual Linux Showcase and Conference (ALS 2000) - USENIX*, vol. 2, pp. 305–316, 2000.

[67] F. Ferreira, N. Santos, and M. Antunes, "Clusters de alta disponibilidade – uma abordagem Open Source," *Actas da conferência Engenharias'2005 - UBI, Covilhã, Portugal*, 2005.

[68] F. Haas, "The Linux-HA User's Guide," http://www.linux-ha.org/doc/ (acedido em Junho de 2010), Edition 0.9.0 (2009).

[69] M. Antunes, *Nova Plataforma Operacional - Gestão de entrepostos.*: Tecnical Report: Sonae Redes de Dados & ISEP-Instituto Politécnico do Porto, 1999.

[70] M. Levy and B. Weitz, *Retailing Management*, 7th ed: McGraw-Hill/Irwin; ISBN: 978-0073381046, 2008.

[71] D. W. Fogarty, J. H. Blackstone, and T. R. Hoffmann, *Production & inventory management*: South-Western Pub. Co.; ISBN: 0538074612 1991.

[72] D. Waters, *Inventory control and management*: A1bazaar; ISBN: 978-0470858769, 2008.

[73] P. Romualdo, "Implementação de um sistema de logística integrada," Universidade do Minho - Relatório de estágio da Licenciatura em Engenharia Informática, 1997.

[74] J. Fernie and L. Sparks, *Logistics and Retail Management: Emerging Issues and New Challenges in the Retail Supply Chain*, 3rd ed: Kogan Page; ISBN: 978-0749454074, 2009.

[75] J. C. Alves, "Backups Centralizados com ADSM (Adstar Distributed Storage Manager)," Universidade do Minho - Relatório de estágio da Licenciatura em Engenharia Informática, 1999.

[76] C. Brooks, P. McFarlane, N. Pott, M. Trcka, and E. Tomaz, *IBM Tivoli Storage Management Concepts*: IBM-SG24-4877-04; ISBN-0738497029; IBM Redbooks, 2007.

[77] Sonae, "Sonae - Áreas de negócio," http://www.sonae.pt/pt/investidores/areas-de-negocio/ (acedido em Maio de 2010).

[78] Sonae, *Annual Report - Consolidated and Individual Accounts - 2008*: Disponível em http://www.sonaeindustria.com/ (acedido em Maio de 2010).

[79] "Microsoft BizTalk Server," http://www.microsoft.com/biztalk (Acedido em Agosto de 2010).

[80] V. Costa, D. Serafim, M. Antunes, and N. Veiga, "IPv6@ESTG-Leiria - Instalação de uma rede piloto," *Engenharias'2005 - Inovação e desenvolvimento; UBI-Covilhã*, pp. 184-188, 2005.

[81] M. Antunes, "Administração centralizada de grupos de sistemas Unix," Universidade do Porto - Faculdade de Ciências; Tese de mestrado em Informática - área de sistemas e redes., 2001.

[82] M. Antunes, "MetaWebmin - Administração centralizada de grupos de sistemas Unix," presented at 4ª Conferência de Redes de Computadores (CRC'2001), 2001.

ANEXOS

- **Anexo A:** Configuração final do cluster de alta disponibilidade implementado para os projectos TEDI e GI.

- **Anexo B:** Conjunto de utilitários para impressão formatada, usados pelos restantes *scripts* desenvolvidos e apresentados em anexo.

- **Anexo C:** *Script* desenvolvido para notificar automaticamente paragens das aplicações.

- **Anexo D:** *Scripts* desenvolvidos para controlar a operacionalidade da base de dados Oracle.

- **Anexo E:** *Scripts* de arranque e paragem das aplicações controladas pelo cluster de HACMP do projecto TEDI.

- **Anexo F:** Descrição sumária dos eventos mais importantes definidos por omissão no HACMP.

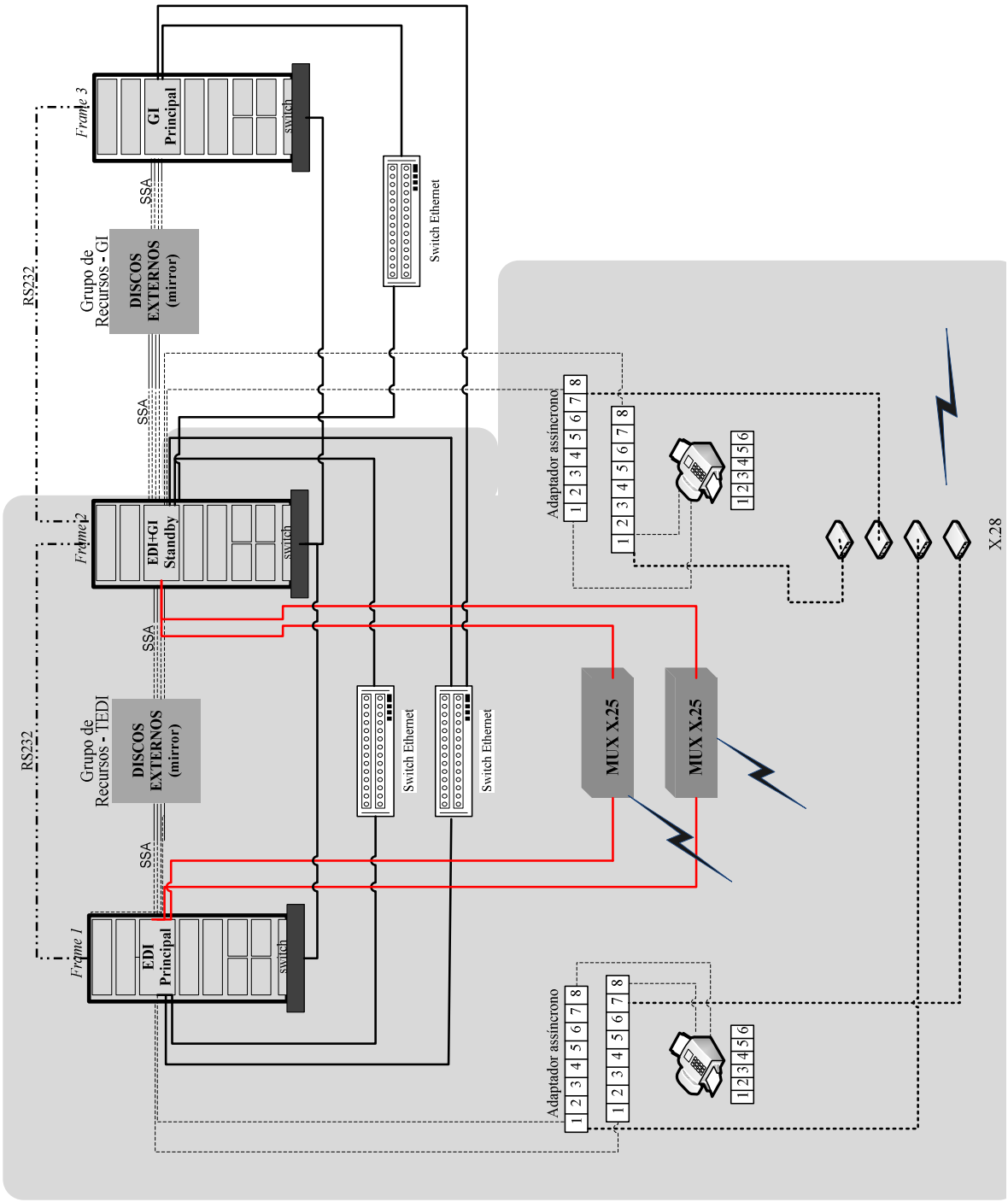
- **Anexo G:** Extracto do ficheiro `/tmp/hacmp.out` relativo à paragem do HACMP no nó principal com retoma dos recursos pelo nó de *standby*.

- **Anexo H:** Guião de configuração de um cluster de alta disponibilidade.

- **Anexo I:** Ficheiro `/etc/inittab` comentado, com observações às entradas específicas do HACMP.

- **Anexo J:** Esquema geral da arquitectura do cluster HACMP de alta disponibilidade do projecto NPO.

ANEXO A


Frame 3
GI Principal
switch
SSA
Grupo de Recursos - GI
DISCOS EXTERNOS (mirror)
RS232
Switch Ethernet
Frame 2
EDI+GI Standby
Grupo de Recursos - TEDI
DISCOS EXTERNOS (mirror)
Frame 1
EDI Principal
MUX X.25
MUX X.25
Adaptador assíncrono
1 2 3 4 5 6 7 8
1 2 3 4 5 6
X.28

ANEXO B

O *script* apresentado de seguida contém um conjunto de utilitários para impressão formatada.

```
# hautil.ksh
#
# Funções para incluir noutros scripts, com . /usr/ha/bin/hautil.ksh
#Contém diversas funções de formatação úteis para incluir nos
#diversos scripts para registo nos ficheiros de log.
#
PROGNAME=`basename $0`
LOG=/usr/ha/logs/$PROGNAME.log

msg()
{
        echo "$*"
}
msgdate()
{
        echo "`date +'%d-%m-%y %H:%M:%S'`: $*"
}
msgprog()
{
        echo "$PROGNAME: $*"
}
message()
{
        echo "$PROGNAME `date +'%d-%m-%y %H:%M:%S'`: $*"
}
msglog()
{
        echo "$*" | tee -a $LOG
}

msgdatelog()
{
        msgdate "$*" | tee -a $LOG
}

msgproglog()
{
        msgprog "$*" | tee -a $LOG
}

messagelog()
{
        message "$*" | tee -a $LOG
}
```

ANEXO C

Neste anexo é apresentado um *script* associado à notificação automática de eventos. Foi definido um *script* onde são efectuados os testes relativos às aplicações que estão a executar no cluster. O procedimento consistia em testar a operacionalidade do servidor Oracle e notificar a equipa de gestão da SRD.

```
#!/usr/bin/ksh
#####################################################
#
# Script com procedimentos activados por erros de software
# no Cluster CLEDI.
#  Só é executado quando acontecer um erro de Software do tipo
#permanente.
#
# Parâmetros
#
# $1 - Sequence number
# $2 - Error ID
# $3 - Error class
# $4 - Error Type
# $5 - Alert Flag Values
# $6 - Resource Name
# $7 - Resource Type
# $8 - Resource Class
#
#####################################################
. /usr/ha/bin/hautil.ksh

msgdatelog "Starting execution"
msgdatelog "Arguments: $*"

SEQUENCE_NUMBER=$1
RESOURCE_NAME=$6

# Definição de variável para envio de mail ao responsável da aplicação

MAIL_ADDR=<numero_bip>@pager.telechamada.pt

# Recolha do nome da aplicação no ficheiro errlog

ERROR_MSG=`errpt -al $SEQUENCE_NUMBER | tail -1`
APP_NAME=`echo $ERROR_MSG | cut -d- -f1`
msgdatelog "ERROR_MSG = \"$ERROR_MSG\""

# Função do envio de beep

Envia_Bip ()
{
   mail $MAIL_ADDR <<EOF
```

```
Node: `hostname`
$ERROR_MSG
EOF
}

# Teste do servidor ORACLE. O código ORACLE está a ser inserido no
#procedimento de teste de operacionalidade.
# Todos os outros testes relativos a aplicações deverão ser colocado
#aqui.

if [ $APP_NAME = "Oracle" ]
then
   msgdatelog "Message is $ERROR_MSG"
   Envia_Bip
   messagelog "Mail sent to $MAIL_ADDR"
fi
```

ANEXO D

São apresentados de seguida os *scripts* desenvolvidos para controlar a operacionalidade da base de dados Oracle. No caso de inoperacionalidade da base de dados era enviada uma mensagem formatada para o ficheiro de *log* do sistema (`/var/adm/errlog`).

```
#!/usr/bin/ksh
#
# oracle.check
#
#Este procedimento deverá estar submetido no cron para ser executado #de 5 em 5
minutos.
#Verifica:
#     - a existência de todos os processos ORACLE
#     - testa a ligação ao Oracle
#
# Argumentos:
#   $1 - Oracle User
#   $2 - Oracle Instance (ORACLE_SID)
#
#   99 - Erro
#

. /usr/ha/bin/hautil.ksh

USAGE="$PROGNAME <oracle_user> <oracle_sid>"

#-----------------------------------------------------------------------
# Verifica argumentos
#

if [ $# -ne 2 ]
then
      echo "Número de parâmetros inválido."
      echo "$USAGE"
      exit 99
fi

ORACLE_USER=$1
ORACLE_SID=$2

# Executa a script oracle.verifica.
# Deverá ser chamado com dois parâmetros:
#     - Utilizador Oracle
#     - Nome da instância a testar.
#No caso de haver várias instâncias, deverá ser executado o
#procedimento para cada uma das instâncias.

/usr/ha/utils/oracle.verifica $ORACLE_USER $ORACLE_SID

# Return Code da execução do procedimento oracle.verify
RC=$?

# Nesta instrução de case são tomadas acções de acordo com o código #retornado
da execução do procedimento oracle.verifica
```

```
case $RC in
    0) msgdatelog "Oracle activo !..."
       #Oracle está activo. Verifica se há ligação a uma instância.

       msgdatelog "Verifica ligação ao Oracle..."

       # Este script SQL, oracle-check.sql, efectua uma operação de            #
SELECT ao DUAL. Uma das muitas possibilidades de verificar a                    #
ligação ao Oracle. Considerei esta apenas por simplicidade de                   #
implementação e fiabilidade.

       ORACLE_MSG=`su - oracle -c \
          "ORACLE_SID=$ORACLE_SID svrmgrl < \
          /usr/ha/oracle/oracle check.sql |grep '^...-.....:'"`

        if [ -z "$ORACLE_MSG" ]
        then
             msgdatelog "Teste de ligação efectuado com sucesso."
        else
             msgdatelog "Teste de ligação falhou. Efectua Notificação..."

             # Envia para o ficheiro errlog uma notificação do Erro.
             # Quando o daemon de registo dos erros receber esta
             # mensagem evocará o script que lhe está associado.

             errlogger "Oracle-ERRO. Instancia $ORACLE_SID. `echo \
              $ORACLE_MSG | tail -1`"
        fi

        break
        ;;

    1) msgdatelog "Instâncias Oracle em baixo !..."

       # Neste ponto faz sentido arrancar automaticamente o Oracle.
       # Por enquanto, não está a ser efectuada nenhuma acção, apenas o    #
       resgisto no ficheiro de log.

       break
      ;;

    2) msgdatelog "Alguns processos Oracle Parados. Notificação..."

       # Processo de notificação idêntico ao efectuado anteriormente.

       errlogger "Oracle-Instancia $ORACLE_SID em baixo"
       break
       ;;

    3) msgdatelog "Oracle em baixo, algumas ligações activas."

       # Não há nenhum processo de Oracle activo. No entanto, existem          #
algumas ligações à Base de Dados. Estes processos estão                         #
filhos do init, ou seja, o processo que os lançou já morreu,                    #     e
deverá efectuar-se o kill de cada um deles.
      # Não está também a ser efectuada qualquer acção.

      break
      ;;
```

```
   *) break
      ;;
esac
```

Este *script* é executado pelo `oracle.check`, apresentado no início deste anexo. Verifica se o sistema gestor de base de dados Oracle está em execução. As validações realizadas são as seguintes:

- Estado dos processos de Oracle.
- Ligações pendentes ao Oracle, estando a instância em baixo.
- Verificar se alguns processos de Oracle estão inactivos.
- Verificar se há segmentos de memória alocados com processos Oracle.

```
#!/usr/bin/ksh
#
# oracle.verifica
#
# Argumentos:
#     $1 - Utilizador Oracle
#     $2 - Instância Oracle (ORACLE_SID)
#
# Return codes (RC) - A devolver ao script oracle.check.
#     0-Instância Oracle activa
#     1-Instância Oracle inactiva (sem processos em execução).
#     2-Alguns processos relativos à instância não existem no sistema
#     3-Instância inactiva. Existência de processos zombie ligados.
#     99-Erro
#
PROGNAME=`basename $0`
USAGE="$PROGNAME <oracle_user> <oracle_sid>"

#Define o inicio da string relativo aos processos que é
#Imprescindivel estarem em execução no sistema. O resto da string #contém o
nome da instância.
#Assim, para a instância EDI, os processos de Oracle são:
#     ora_smon_EDI
#     ora_lgwr_EDI
#     ora_reco_EDI
#     ora_pmon_EDI
#     ora_dbwr_EDI

INSTANCE_PROCS="ora_smon_ ora_lgwr_ ora_reco_ ora_pmon_ ora_dbwr_"

#Número total dos processos Oracle. Deverá coincidir com o número de #processos
definidos na variável INSTANCE_PROCS.
TOT_INST_PROCS=5

#-----------------------------------------------------------------------
# Verifica argumentos

if [ $# -ne 2 ]
then
        echo "Número de argumentos inválido."
        echo "$USAGE"
        exit 99
```

```
fi

ORACLE_USER=$1
ORACLE_SID=$2

STATUS=0

#-------------------------------------------------------------------------
# Verifca se todos os processos da instância estão em execução.
# Se for verdade, considera-se que a instância está activa.

GREP_EXP=`echo $INSTANCE_PROCS | sed "s/_ /_$ORACLE_SID|/g"`$ORACLE_SID
GREP_CAB="UID   PID  PPID   C    STIME    TTY  TIME CMD"

# Calcula número de processo da instância em execução no sistema.

NUM_INST_PROCS=`ps -ef | grep -wE "$GREP_EXP" | grep -v grep | \
                grep -v "$GREP_CAB" | wc -l`

# Se o numero de processos não for igual ao definido !...

if [ $NUM_INST_PROCS -ne $TOT_INST_PROCS ]
then
      NUM_ORA_PROCS=`ps -fu $ORACLE_USER | grep -w oracle$ORACLE_SID |
          grep -v grep | grep -v "$GREP_CAB" | wc -l`

      # se não existe qualquer processo em execução

      if [ $NUM_INST_PROCS -eq 0 ]
      then
            # Verifica se existem ligações à instância Oracle
            # mesmo estando a instância em baixo.

            if [ $NUM_ORA_PROCS -eq 0 ]
                  then
                        # Testa segmentos de memória alocada por
                        # ligações Oracle que ainda existam.

                        IPC=`ipcs | grep oracle | wc -l`
                        if [ $IPC -eq 0 ]
                        then
                              STATUS=1
                              # Instância em baixo !...
                        else
                              STATUS=2
                              # Instância em baixo mas com segmentos
                              # de memória alocados por ligações à
                              # base de dados Oracle.
                        fi
            else
                  STATUS=3
                  # Instância em baixo com ligações ao Oracle.
            fi

      else
            STATUS=2
            # Instância com alguns processo activos e outros não.
            # A necessitar de intervenção.
      fi
fi
```

```
exit $STATUS    # Retorna para script oracle.check o valor da variável $STATUS.
```

O *script* `oracle.clean` tem por objectivo remover do sistema os processos pendentes de ligação à base de dados Oracle, quando esta já não se encontra activa.

```
#!/usr/bin/ksh
#
# oracle.clean
#
# Argumentos:
#     $1 - Utilizador Oracle
#     $2 - Instância Oracle (ORACLE_SID)
#
# Código retornado :
#     0 - OK
#     1 - ERROR
#

. /usr/ha/bin/hautil.ksh

USAGE="$PROGNAME <oracle_user> <oracle_sid>"
INSTANCE_PROCS="ora_smon_ ora_lgwr_ ora_snp0_ ora_reco_ ora_pmon_ ora_dbwr_"
TOT_INST_PROCS=6
#-------------------------------------------------------------------------
# Verifica argumentos
#

if [ $# -ne 2 ]
then
        echo "Numero de argumentos inválido."
        echo "$USAGE"
        exit 99
fi

ORACLE_USER=$1
ORACLE_SID=$2

STATUS=0

msgdatelog "Início de execução."

#-------------------------------------------------------------------------
#A instância Oracle será parada em modo ABORT. Deve apenas ser executado
#quando, alguns processos da base de dados não estão activos.

su - oracle -c "ORACLE_SID=$ORACLE_SID svrmgrl \
     < /usr/ha/oracle/shut-abort.sql" | tee -a $LOG

GREP_CAB="UID   PID  PPID   C    STIME    TTY  TIME CMD"

for i in 15 1 9
do
      ORA_PROCS=`ps -ef | grep -w oracle$ORACLE_SID |
      grep -v grep | grep -v "$GREP_CAB" | awk '{print $2]'`

      msgdatelog "Existem processos activos ao Oracle: $ORA_PROCS"

      if [ -n "$ORA_PROCS" ]
      then
```

```
            msgdatelog "Espera 10 segundos ..."
            sleep 10
            msgdatelog "Tenta com kill -$i ..."
            kill -$i $ORA_PROCS
      else
            break
      fi

done
msgdatelog "FIM."
```

ANEXO E

Os *scripts* apresentados neste anexo referem-se ao arranque e paragem das aplicações incluídas no grupo de recursos do cluster de HACMP. Cada aplicação continha um *script* de arranque e paragem específico.

Procedimento de paragem:

```
#!/usr/bin/ksh
#
# EDIAPP.stop
#
# Script para paragem das aplicações do cluster EDI.
# Genérico para permitir o arranque de várias aplicações.

. /usr/ha/bin/hautil.ksh

NODE=`hostname | cut -d. -f1`

APPSRV=SRVEDI

msglog "------------------------------------------------------------"
msgdatelog "Paragem do servidor de aplicações: $APPSRV"
msgdatelog "Início --> Node: $NODE"
msglog "------------------------------------------------------------"

msglog "------------------------------------------------------------"
msgdatelog "Paragem do software EDI . Nó: $NODE"
msglog "------------------------------------------------------------"

su - <user> -c "stop_appX.sh"

msglog "------------------------------------------------------------"
msgdatelog "Paragem da aplicação Y. Nó: $NODE"
msglog "------------------------------------------------------------"

su - <user> -c "stop_appY.sh"

msglog "------------------------------------------------------------"
msgdatelog "Paragem de Listener SQL*Net:  Nó: $NODE"
msglog "------------------------------------------------------------"

/usr/bin/su - oracle -c "ORACLE_SID=TEDI lsnrctl stop"

msglog "------------------------------------------------------------"
msgdatelog "Paragem de Oracle Server. Nó: $NODE"
msglog "------------------------------------------------------------"

/usr/bin/su - oracle -c \
"ORACLE_SID=TEDI svrmgrl < /usr/ha/oracle/shut-immed.sql"

msglog "------------------------------------------------------------"
msgdatelog "Limpeza de processos Oracle pendentes. Node: $NODE"

msglog "------------------------------------------------------------"

./oracle.verifica oracle EDI
RC=$?
```

```
case $RC in
  0) msgdatelog "Oracle activo !!! Processo de limpeza..."
      # Procedimento que efectua o kill de ligações existentes ao
      # Oracle após a paragem do servidor.

      ./oracle.clean oracle EDI
      break
      ;;

  1) msgdatelog "Oracle em baixo."
     break
     ;;

  2|3) msgdatelog "Oracle parcialmente em baixo. Processo de limpeza..."
       ./oracle.clean oracle EDI
       break
       ;;

  *)  msgdatelog "Código de retorno inválido !"
      break
      ;;
esac

msglog "-------------------------------------------------------------"
msgdatelog "Servidor de aplicações STOP: $APPSRV"
msgdatelog "FIM. Nó: $NODE"
msglog "-------------------------------------------------------------"
```

Procedimento de arranque.

```
#!/usr/bin/ksh
#
# EDIAPP.start
#
# Script de arranque do servidor de aplicações

. /usr/ha/bin/hautil.ksh

NODE=`hostname | cut -d. -f1`
APPSRV=SRVED
ORACLE_SID=XEDI

msglog "-------------------------------------------------------------"
msgdatelog "Servidor de Aplicações START: $APPSRV"
msgdatelog "Início: Nó: $NODE"
msglog "-------------------------------------------------------------"

msglog "-------------------------------------------------------------"
msgdatelog "Permissões de Datafiles Oracle: Instância $ORACLE_SID.Node: $NODE"
msglog "-------------------------------------------------------------"

#Durante o takeover o comando importvg coloca a permissão dos raw devices onde
#estão os datafiles do Oracle com permissões de root.
#Assim, é necessário alterar as permissões durante o arranque doa aplicações.

chown oracle.dba /dev/redi_*
chmod 600 /dev/redi_*

msglog "-------------------------------------------------------------"
```

```
msgdatelog "Arranque de instância Oracle : $ORACLE_SID. Nó: $NODE"
msglog "------------------------------------------------------------"

/usr/bin/su - oracle -c \
      "ORACLE_SID=$ORACLE_SID svrmgrl < /usr/ha/oracle/startup.sql"

msglog "------------------------------------------------------------"
msgdatelog "Verifica operacionalidade do servidor Oracle.  Nó: $NODE"
msglog "------------------------------------------------------------"

/usr/ha/appsrv/oracle.verifica oracle $ORACLE_SID
RC=$?

case $RC in
  0)  msgdatelog "Oracle activo. Procedimento de teste de ligação."

      ORACLE_MSG=`su - oracle -c \
          "ORACLE_SID=$ORACLE_SID svrmgrl < /usr/ha/oracle/oracle
      check.sql | grep '^...-.....:'"`
      if [ -z "$ORACLE_MSG" ]
      then
            msgdatelog "Teste de ligação com sucesso"
      else
            msgdatelog "Teste de ligação falhou"
            msgdatelog "$ORACLE_MSG. Notificação..."

            # Notificação para o errlog do erro de ligação ao
            # Oracle.

            errlogger "Oracle-Erro de ligacao. Instancia $ORACLE_SID.\
                        `echo $ORACLE_MSG | tail -1`"
      fi

      break
      ;;

  1|2|3) msgdatelog "Oracle em baixo. Notificação..."
         # Notificação para o errlog do erro de ligação ao Oracle.
         # errlogger "Oracle-Erro no arranque. Instancia $ORACLE_SID."

         break
         ;;

  *) msgdatelog "Return Code inválido"
     break
     ;;
esac

msglog "------------------------------------------------------------"
msgdatelog "Arranque de SQL*Net Listener: $ORACLE_SID. Nó: $NODE"
msglog "------------------------------------------------------------"

su - oracle -c "ORACLE_SID=$ORACLE_SID lsnrctl start"

msglog "------------------------------------------------------------"
msgdatelog "Arranque do software EDI. Nó: $NODE"
msglog "------------------------------------------------------------"

su - <user> -c "start_appX.sh"

msglog "------------------------------------------------------------"
```

```
msgdatelog "Arranque da aplicação Y. Nó: $NODE"
msglog "------------------------------------------------------------"

su - <user> -c "start_appY.sh"

msglog "------------------------------------------------------------"
msgdatelog "Servidor de aplicações START: $APPSRV"
msgdatelog "FIM. Nó: $NODE"
msglog "------------------------------------------------------------"
```

ANEXO F

Eventos relativos ao nó

1) *node_up***:** Normalmente iniciado por um nó que se está a juntar ao cluster, aquando do arranque do HACMP. É a seguinte a sequência de eventos associada:
    - *node_up_local:* Adquire os endereços IP de serviço do cluster, bem como os seus recursos associados. Este evento é subdividido nos seguintes:
        i. *acquire_service_addr* (quando configurado para *takeover* de IP). Configura o endereço de *boot* para o correspondente de serviço. Executa `telinit -a`. O HACMP altera o ficheiro `/etc/inittab`, alterando o *run level* dos processos TCP/IP para "`a`".
        ii. *acquire_takeover_addr* Verifica se existe um endereço de *standby* configurado e realiza a troca do endereço de *standby* pelo da interface de rede que está a ser desactivada.
        iii. *get_disk_vg_fs*: Assume o controlo de recursos de discos, nomeadamente *volume groups* e *filesystems*.
2) *node_up_local_complete.* Ocorre após o evento *node_up_local* ter concluído com sucesso.
3) *node_down.* Sempre que, intencionalmente, um nó falha e deixa de pertencer ao cluster.
    - *node_down_local***.** Este evento é subdividido nos seguintes:
        i. *stop_server.* Pára as aplicações do cluster.
        ii. *release_takeover_address* (quando configurado para *takeover* de IP). Desactiva o endereço IP da interface de rede.
        iii. *release_vg_fs.* Liberta os *volume groups* e *filesystems* do grupo de recursos.
        iv. *release_service_address***.** Desactiva o endereço de serviço e reconfigura o interface de serviço com o endereço de *boot*.
4) *node_down_local_complete.* Ocorre após o evento *node_down_local* ter concluído com sucesso.

Eventos relativos à rede

1) *network_down***.** Ocorre quando o Cluster Manager determina que a rede falhou. Pode ter duas formas:
    - *network_down local.* Apenas um nó do cluster perde o contacto com a rede.
    - *network_down global.* Todos os nós do cluster perdem o contacto com a rede.

2) *network_down_complete*. Ocorre após o evento *network_down* ter concluído com sucesso.
3) *network_up***.** Ocorre quando o Cluster Manager determina que a rede está disponível.
4) *network_up_complete.* Ocorre após o evento *network_up* ter concluído com sucesso.

Eventos relativos a interfaces de rede

1) *swap_adapter.* Ocorre quando uma interface de serviço num determinado nó falha, trocando ou alterando o endereço IP das interfaces de um nó do cluster.
2) *swap_adapter_complete***.** Ocorre após o evento *swap_adapter* ter concluído com sucesso. Assegura o refrescamento das tabelas de ARP locais, apagando entradas obsoletas e realizando `ping` aos endereços IP dos nós.
3) *fail_standby.* Ocorre quando a interface de *standby* falha ou fica indisponível, devido a um *takeover* de endereço IP.
4) *join_standby.* Este evento ocorre quando a interface de *standby* fica disponível.

ANEXO G

Ficheiro `/tmp/hacmp.out` no nó principal:

```
Apr 18 15:27:58 EVENT START: node_down nodeA gracefull

PRE_EVENT_MEMBERSHIP =nodeA nodeB
POST_EVENT_MEMBERSHIP=nodeB
PRE_EVENT_MEMBERSHIP =nodeA nodeB
POST_EVENT_MEMBERSHIP=nodeB

Apr 18 15:28:00 EVENT START: node_down_local
Apr 18 15:28:00 EVENT START: stop_server SRVEDI

------------------------------------------------------------
18-04-97 15:28:01: Application Server STOP: SRVEDI
18-04-97 15:28:01: Starting execution. Node: nodeA
------------------------------------------------------------
------------------------------------------------------------
18-04-97 15:28:01: Paragem de aplicacao EDI. Node: nodeA
------------------------------------------------------------

#######   Paragem do tradutor EDI   ########
------------------------------------------------------------
18-04-97 15:28:46: Paragem de aplicacao X. Node: nodeA
------------------------------------------------------------

#######   Paragem de aplicacao X    ########

------------------------------------------------------------
18-04-97 15:28:48: Stopping SQL*Net Listener: XEDI. Node: nodeA
------------------------------------------------------------

LSNRCTL for IBM/AIX RISC System/6000: Version 2.3.2.1.0 - Production on
18-APR-97 15:28:49

Copyright (c) Oracle Corporation 1994.  All rights reserved.

Connecting to (ADDRESS=(PROTOCOL=IPC)(KEY=XEDI))
The command completed successfully
------------------------------------------------------------
18-04-97 15:28:50: Stoping Oracle Server. Node: nodeA
------------------------------------------------------------

#######  Paragem de servidor Oracle ########
------------------------------------------------------------
Apr 18 15:29:09 EVENT COMPLETED: stop_server SRVEDI

Apr 18 15:29:09 EVENT START: release_vg_fs /fs_01 /fs_02 vg_01 vg_02

/dev/lv01:   38174c(user2)   39378c(user1)   45756c(user1)   58546c(user1)
73686c (user2)

/dev/lv02:
```

```
Apr 18 15:29:28 EVENT COMPLETED: release_vg_fs /fs_01 /fs_02 vg_01
vg_02


Apr 18 15:29:29 EVENT START: release_service_addr spnodeA

Starting execution of /usr/sbin/cluster/events/utils/cl_nm_nis_off with
parameters:.
Completed execution of /usr/sbin/cluster/events/utils/cl_nm_nis_off
with parameters:. Exit Status = 0.
en1
Starting execution of /usr/sbin/cluster/events/utils/cl_swap_IP_address
with parameters: en1 1.1.1.62 255.255.255.224.
127.0.0.1 net default: gateway 127.0.0.1
127.0.0.1 net 1.1.1.32: gateway 127.0.0.1
/usr/sbin/cluster/events/utils/cl_swap_IP_address: Configuring adapter
en1 at IP address 1.1.1.62
Starting execution of /usr/sbin/cluster/events/utils/cl_swap_HW_address
with parameters: 1.1.1.62 en1.
ent1 Defined
ent1 Available
1.1.1.33 net default: gateway 1.1.1.33
1.1.1.62 net 1.1.1.32: gateway 1.1.1.62
127.0.0.1 net 127: gateway 127.0.0.1
Entry 10.10.0.18 (10.10.0.18) was deleted from local arp table.
Entry 1.1.1.33 (1.1.1.33) was deleted from local arp table.
Entry 1.1.1.35 (1.1.1.35) was deleted from local arp table.
Entry 10.10.0.100 (10.10.0.100) was deleted from local arp table.
Entry 10.10.0.1 (10.10.0.1) was deleted from local arp table.
Entry 10.10.0.30 (10.10.0.30) was deleted from local arp table.
Entry 1.1.1.67 (1.1.1.67) was deleted from local arp table.
Entry 10.10.0.6 (130.130.0.6) was deleted from local arp table.
Entry 10.10.0.10 (10.10.0.10) was deleted from local arp table.
Completed execution of
/usr/sbin/cluster/events/utils/cl_swap_IP_address with parameters: en1
1.1.1.62 255.255.255.224. Exit Status = 0 .
Starting execution of /usr/sbin/cluster/events/utils/cl_nm_nis_on with
parameters:.
Completed execution of /usr/sbin/cluster/events/utils/cl_nm_nis_on with
parameters:. Exit Status = 0.
Apr 18 15:29:38 EVENT COMPLETED: release_service_addr spnodeA

Apr 18 15:29:39 EVENT COMPLETED: node_down_local

PRE_EVENT_MEMBERSHIP=nodeA nodeB
POST_EVENT_MEMBERSHIP=nodeB
Apr 18 15:29:39 EVENT COMPLETED: node_down nodeA gracefull

Apr 18 15:29:40 EVENT START: node_down_complete nodeA

node_down_local_complete

Apr 18 15:29:41 EVENT START: node_down_local_complete
Apr 18 15:29:41 EVENT COMPLETED: node_down_local_complete
Apr 18 15:29:41 EVENT COMPLETED: node_down_complete nodeA
```

Ficheiro `/tmp/hacmp.out` no nó de *standby* (após a paragem no nó principal):

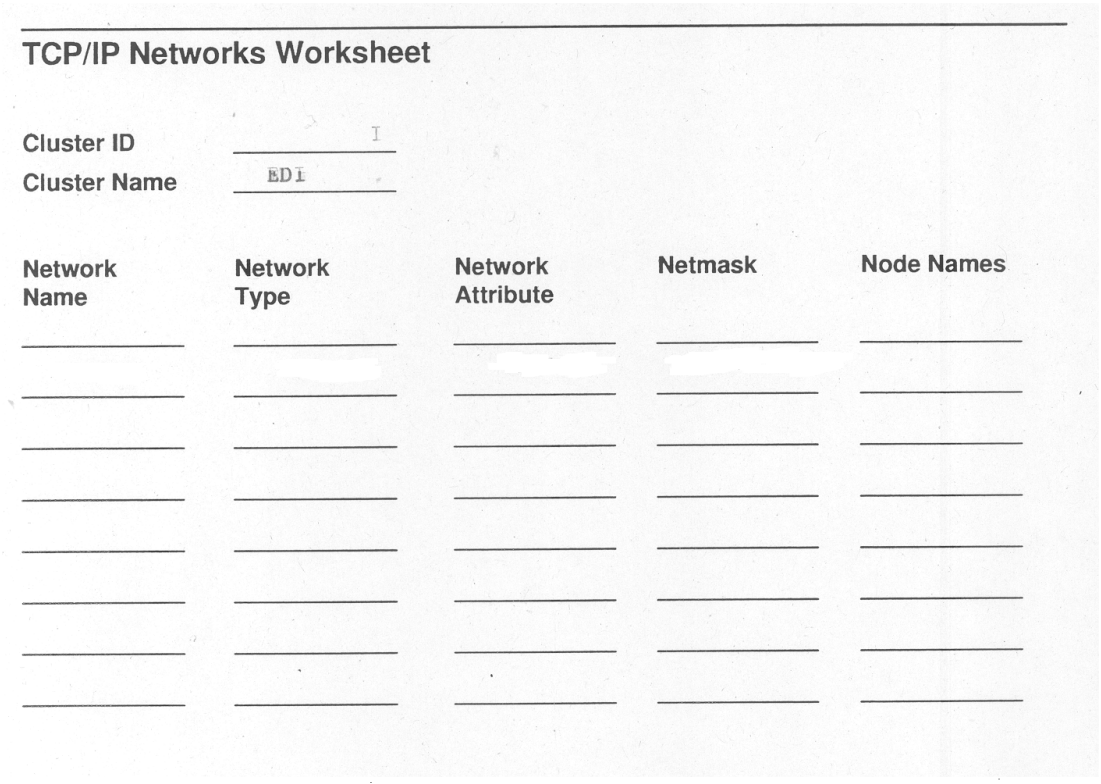
```
Apr 18 15:29:40 EVENT START: node_down nodeA

PRE_EVENT_MEMBERSHIP=nodeA nodeB
POST_EVENT_MEMBERSHIP=nodeB
PRE_EVENT_MEMBERSHIP=nodeA nodeB
POST_EVENT_MEMBERSHIP=nodeB

Apr 18 15:29:41 EVENT START: node_down_remote nodeA


Apr 18 15:29:42 EVENT START: acquire_takeover_addr spnodeA

Starting execution of /usr/sbin/cluster/events/utils/cl_nm_nis_off with
parameters: .
Completed execution of /usr/sbin/cluster/events/utils/cl_nm_nis_off
with parameters: . Exit Status = 0 .
Starting execution of /usr/sbin/cluster/events/utils/cl_swap_IP_address
with parameters: en2 1.1.1.34 255.255.255.224.
127.0.0.1 net 4: gateway 127.0.0.1
/usr/sbin/cluster/events/utils/cl_swap_IP_address: Configuring adapter
en2 at IP address 1.1.1.34
Starting execution of /usr/sbin/cluster/events/utils/cl_swap_HW_address
with parameters: 1.1.1.34 en2.
ent2 Defined
ent2 changed
ent2 Available
ent2 changed
1.1.1.67 net 1.1.1.64: gateway 1.1.1.67
127.0.0.1 net 127: gateway 127.0.0.1
1.1.1.67 net 1.1.1.64: gateway 1.1.1.667
Entry 1.1.1.33 (1.1.1.33) was deleted from local arp table.
Entry 10.10.0.22 (130.130.0.22) was deleted from local arp table.
Entry 1.1.1.34 (1.1.1.34) was deleted from local arp table.
Entry 10.10.0.100 (10.10.0.100) was deleted from local arp table.
Entry 1.1.1.62 (1.1.1.62) was deleted from local arp table.
Entry 10.10.0.1 (10.10.0.1) was deleted from local arp table.
Entry 1.1.1.66 (1.1.1.66) was deleted from local arp table.
Entry 10.10.0.30 (10.10.0.30) was deleted from local arp table.
Entry 10.10.0.6 (10.10.0.6) was deleted from local arp table.
Entry 10.10.0.8 (10.10.0.8) was deleted from local arp table.
Entry 10.10.0.10 (10.10.0.10) was deleted from local arp table.
Entry 10.10.0.14 (10.10.0.14) was deleted from local arp table.
Completed execution of
/usr/sbin/cluster/events/utils/cl_swap_IP_address with parameters: en2
1.1.1.34 255.255.255.224. Exit Status = 0 .
Starting execution of /usr/sbin/cluster/events/utils/cl_nm_nis_on with
parameters: .
Completed execution of /usr/sbin/cluster/events/utils/cl_nm_nis_on with
parameters: . Exit Status = 0 .
Apr 18 15:29:54 EVENT COMPLETED: acquire_takeover_addr spnodeA
```

```
Apr 18 15:29:54 EVENT START: get_disk_vg_fs /fs_01 /fs_02 vg_01 vg_02

Starting execution of /usr/sbin/cluster/events/utils/cl_disk_available
with parameters: hdisk44 hdisk45 hdisk46 hdisk47 .
Completed execution of /usr/sbin/cluster/events/utils/cl_disk_available
with parameters: hdisk44 hdisk45 hdisk46 hdisk47 . Exit Status = 0 .

** Checking /dev/rlv_01 (/fs_01)
** Unmounted cleanly - Check suppressed

Replaying log for /dev/lv_01.

** Checking /dev/rlvgs (/fs_02)
** Unmounted cleanly - Check suppressed
Apr 18 15:30:33 EVENT COMPLETED: get_disk_vg_fs /fs_01 /fs_02 vg_01
vg_02

Apr 18 15:30:34 EVENT COMPLETED: node_down_remote nodeA

PRE_EVENT_MEMBERSHIP=nodeA nodeB
POST_EVENT_MEMBERSHIP=nodeB
Apr 18 15:30:34 EVENT COMPLETED: node_down nodeA


Apr 18 15:30:35 EVENT START: node_down_complete nodeA

node_down_remote_complete

nodeA

Apr 18 15:30:35 EVENT START: node_down_remote_complete nodeA


Apr 18 15:30:36 EVENT START: start_server SRVEDI

------------------------------------------------------------
Apr 18 15:30:36 EVENT COMPLETED: start_server SRVEDI
18-04-97 15:30:37: Application Server START: SRVEDI

18-04-97 15:30:37: Starting execution. Node: nodeB
Apr 18 15:30:37 EVENT START: fail_standby nodeB 1.1.1.67

------------------------------------------------------------
------------------------------------------------------------
Apr 18 15:30:37 EVENT COMPLETED: node_down_remote_complete nodeA

18-04-97 15:30:37: Changing Datafiles permissions: Instance XEDI. Node:
nodeB
------------------------------------------------------------
------------------------------------------------------------
Apr 18 15:30:38 EVENT COMPLETED: fail_standby nodeB 1.1.1.67
18-04-97 15:30:38: Starting Oracle Instance: XEDI. Node: nodeB

Apr 18 15:30:38 EVENT COMPLETED: node_down_complete nodeA

------------------------------------------------------------
```

```
#######  Arranque do Oracle ########

------------------------------------------------------------
18-04-97 15:31:17: Verifying Oracle Server. Node: nodeB
------------------------------------------------------------
18-04-97 15:31:17: Oracle is  UP. Procceding to connection test...
18-04-97 15:31:19: Connection Test Successful.
------------------------------------------------------------
18-04-97 15:31:19: Starting SQL*Net Listener: XEDI. Node: nodeB
------------------------------------------------------------

#######  Arranque do Listener Oracle ######

------------------------------------------------------------
18-04-97 15:31:32: Arranque da aplicacao X. Node: nodeB
------------------------------------------------------------

------------------------------------------------------------
18-04-97 15:31:38: Application Server START: SRVEDI
18-04-97 15:31:38: Terminating. Node: nodeB
------------------------------------------------------------
```

ANEXO H

Este anexo contém uma proposta de guião de configuração de um cluster de alta disponibilidade. Este guião está dividido em vários tópicos, designadamente a configuração da rede, dos discos (física e lógica), das aplicações e dos grupos de recurso.



TCP/IP Networks Worksheet

Cluster ID _____

Cluster Name EDI

Network Name	Network Type	Network Attribute	Netmask	Node Names

TCP/IP Network Adapter Worksheet

Node Name _____

Interface Name	Adapter IP Label	Adapter Function	Adapter IP Address	Network Name	Network Attribute	Adapter HW Address

Serial Networks Worksheet

Cluster ID _____

Cluster Name _____

Network Name	Network Type	Network Attribute	Node Names
_____	_____	serial	_____
_____	_____	serial	_____
_____	_____	serial	_____
_____	_____	serial	_____
_____	_____	serial	_____
_____	_____	serial	_____
_____	_____	serial	_____
_____	_____	serial	_____

Serial Network Adapter Worksheet

Node Name _____

Slot Number	Interface Name	Adapter Label	Network Name	Network Attribute	Adapter Function
_____	_____	_____	_____	serial	service
_____	_____	_____	_____	serial	service
_____	_____	_____	_____	serial	service
_____	_____	_____	_____	serial	service
_____	_____	_____	_____	serial	service
_____	_____	_____	_____	serial	service
_____	_____	_____	_____	serial	service
_____	_____	_____	_____	serial	service

Shared IBM 9333 Serial Disk Worksheet

	Node A	Node B	Node C	Node D
Node Name	____	____	____	____
Slot Number	____	____	____	____
Logical Name	____	____	____	____

IBM 9333 Drawer/Desk Label ____

Adapter I/O Connector	____	____	____	____
Controller Logical Name	____	____	____	____

IBM 9333 Shared Drives in Node Name ____

	Size (MB)	Logical Device Name		
Drive (1)	( )____	____	____	____
Drive (2)	( )____	____	____	____
Drive (3)	( )____	____	____	____
Drive (4)	( )____	____	____	____

IBM 9333 Drawer/Desk Label ____

Adapter I/O Connector	____	____	____	____
Controller Logical Name	____	____	____	____

IBM 9333 Shared Drives in Node Name ____

	Size (MB)	Logical Device Name		
Drive (1)	( )____	____	____	____
Drive (2)	( )____	____	____	____
Drive (3)	( )____	____	____	____
Drive (4)	( )____	____	____	____

IBM 9333 Drawer/Desk Label ____

Adapter I/O Connector	____	____	____	____
Controller Logical Name	____	____	____	____

IBM 9333 Shared Drives in Node Name ____

	Size (MB)	Logical Device Name		
Drive (1)	( )____	____	____	____
Drive (2)	( )____	____	____	____
Drive (3)	( )____	____	____	____
Drive (4)	( )____	____	____	____

Shared IBM Serial Storage Architecture Disk Subsystems Worksheet

Host and Adapter Information

	Node A	Node B	Node C	Node D
Node Name	_____	_____	_____	_____
SSA Adapter Label	_____	_____	_____	_____
Slot Number	_____	_____	_____	_____
Dual-port Number	_____	_____	_____	_____

SSA Logical Disk Drive

Logical Device Name

Node A	Node B	Node C	Node D
_____	_____	_____	_____
_____	_____	_____	_____
_____	_____	_____	_____
_____	_____	_____	_____
_____	_____	_____	_____

SSA Logical Disk Drive

Logical Device Name

Node A	Node B	Node C	Node D
_____	_____	_____	_____
_____	_____	_____	_____
_____	_____	_____	_____
_____	_____	_____	_____
_____	_____	_____	_____

Non-Shared Volume Group Worksheet (e.g. rootvg)

Node Name _____

Volume group name _____
Physical volumes _____
_____
_____

Logical volume name _____
Number of copies of logical partition _____
On separate physical volumes? _____
File system mount point _____
Size (512-byte) _____

Logical volume name _____
Number of copies of logical partition _____
On separate physical volumes? _____
File system mount point _____
Size (512-byte) _____

Shared Volume Group/File System Worksheet (Non-Concurrent Access)

	Node A	Node B	Node C	Node D
Node Names				
Shared volume group name				
Major number				
Log logical volume name				
Physical volumes				

Logical volume name	
Number of copies of logical partition	
On separate physical volumes?	
File system mount point	
Size (512-byte)	

Logical volume name	
Number of copies of logical partition	
On separate physical volumes?	
File system mount point	
Size (512-byte)	

Logical volume name	
Number of copies of logical partition	
On separate physical volumes?	
File system mount point	
Size (512-byte)	

Application Server Worksheet

Cluster ID _____

Cluster Name _____

Note: Use full path names for all user-defined scripts.

Application _____

Server name _____

Start Script _____

Stop Script _____

Application _____

Server name _____

Start Script _____

Stop Script _____

Application _____

Server name _____

Start Script _____

Stop Script _____

Application _____

Server name _____

Start Script _____

Stop Script _____

Application Worksheet

Application Name: _____

	Directory/Path	File System	Location	Sharing
Executables:	_____	_____	_____	_____
Configuration Files:	_____	_____	_____	_____
Data Files/Devices:	_____	_____	_____	_____
Log Files/Devices:	_____	_____	_____	_____

Cluster Name: _____

Node Relationship: _____

Fallover Strategy:

Nodes: _____ _____ _____ _____ _____ _____ _____ _____

Strategy: _____ _____ _____ _____ _____ _____ _____ _____

Normal Start Commands/Procedures:

_____

_____

_____

_____

Verification Commands/Procedures:

_____

_____

_____

_____

Node Reintegration/Takeover Caveats:

Node: _____ Caveats: _____

Node: _____ Caveats: _____

Node: _____ Caveats: _____

Node: _____ Caveats: _____

Node: _____ Caveats: _____

Node: _____ Caveats: _____

Node: _____ Caveats: _____

Node: _____ Caveats: _____

Resource Group Worksheet

Cluster ID _____

Cluster Name _____

Resource Group Name _____

Node Relationship _____

Participating Node Names _____

Service IP Label _____

Filesystems _____

Filesystems to Export _____

Filesystems to NFS Mount _____

Volume Groups _____

Raw Disks _____

Application Servers _____

Inactive Takeover _____

Resource Group Name _____

Node Relationship _____

Participating Node Names _____

Service IP Label _____

Filesystems _____

Filesystems to Export _____

Filesystems to NFS Mount _____

Volume Groups _____

Raw Disks _____

Application Servers _____

Inactive Takeover _____

Cluster Event Worksheet

Note: Use full path names for all user-defined scripts.

Cluster ID _____

Cluster Name _____

Cluster Event Name _____

Event Command _____

Notify Command _____

Pre-Event Command _____

Post-Event Command _____

Event Recovery Command _____

Recovery Counter _____

Cluster Event Name _____

Event Command _____

Notify Command _____

Pre-Event Command _____

Post-Event Command _____

Event Recovery Command _____

Recovery Counter _____

Cluster Event Name _____

Event Command _____

Notify Command _____

Pre-Event Command _____

Post-Event Command _____

Event Recovery Command _____

Recovery Counter _____

ANEXO I

Este anexo apresenta o ficheiro `/etc/inittab` num nó do cluster, com as entradas relativas ao HACMP.

```
init:2:initdefault:
brc::sysinit:/sbin/rc.boot 3 >/dev/console 2>&1 # Phase 3 of system boot
start_net:2:wait:/usr/lpp/ssp/install/bin/start_net > /dev/console
powerfail::powerfail:/etc/rc.powerfail 2>&1 | alog -tboot > /dev/console
(...)
rc:2:wait:/etc/rc 2>&1 | alog -tboot > /dev/console # Multi-User checks
srcmstr:2:respawn:/usr/sbin/srcmstr # System Resource Controller

#
# Arranque da configuração de rede para o HACMP.
#

harc:2:wait:/usr/sbin/cluster/etc/harc.net # HACMP for AIX network startup
(...)

#O runlevel corresponde ao segundo campo do ficheiro. Cada campo contém o
#separador ´:´.
#

rctcpip:a:wait:/etc/rc.tcpip > /dev/console 2>&1 # Start TCP/IP daemons
(...)
qdaemon:a:wait:/usr/bin/startsrc -sqdaemon
writesrv:a:wait:/usr/bin/startsrc -swritesrv
uprintfd:2:respawn:/usr/sbin/uprintfd
rcncs:a:wait:sh /etc/rc.ncs > /dev/console 2>&1 # Start NCS
jm_setupcss:2:boot:/usr/lpp/ssp/bin/jm_setupcss
browser:a:respawn:/usr/lpp/xlC/browser/pdnsd
sysctld:a:once:/usr/bin/startsrc -s sysctld
tty6:2:off:/usr/sbin/getty /dev/tty6
tty9:2:off:/usr/sbin/getty /dev/tty9

#
# Arranque do HACMP e serviços associados.
#
#Após o arranque dos serviços de TCP/IP e de rede para o HACMP, a entrada
#seguinte disponibiliza o switch do SP2 para o nó em causa.

unfe:a:once:/usr/ha/bin/haunfence >/dev/console 2>&1
clinit:a:wait:touch /usr/sbin/cluster/.telinit
# HACMP for AIX This must be last entry in inittab!
#
# Activa o software do cluster HACMP.
hacmp6000:2:wait:/usr/sbin/cluster/etc/rc.cluster -boot  -i  # Bring up Cluster
#
# Deverá haver o máximo de cuidado com os runlevel de cada uma das entradas.
# Assim, primeiro o arranque dos serviços de rede de TCP/IP, depois os de HACMP
# e, no fim de tudo, o HACMP.
```

ANEXO J

Este anexo contempla o esquema com a topologia do cluster de HACMP, instalado em cada entreposto no projecto NPO-Entrepostos.