



UNIVERSIDADE AUTÓNOMA DE LISBOA

DEPARTAMENTO DE CIÊNCIAS ECONÓMICAS, EMPRESARIAIS
E TECNOLÓGICAS

MESTRADO EM ENGENHARIA E TECNOLOGIAS INFORMÁTICAS

***Criptosistema para sistema de ficheiros de grids de
armazenamento***

Relatório de Actividade Profissional para a obtenção do grau de Mestre em Engenharia e Tecnologias
Informáticas

Mestrando: Francisco José Fialho Nunes

Orientador: Professor Doutor *Alberto Carneiro*

Lisboa, Maio 2014



UNIVERSIDADE AUTÓNOMA DE LISBOA

DEPARTAMENTO DE CIÊNCIAS ECONÓMICAS, EMPRESARIAIS
E TECNOLÓGICAS

MESTRADO EM ENGENHARIA E TECNOLOGIAS INFORMÁTICAS

***Criptosistema para sistema de ficheiros de grids de
armazenamento***

Relatório de Actividade Profissional para a obtenção do grau de Mestre em Engenharia e Tecnologias
Informáticas

Mestrando: Francisco José Fialho Nunes

Júri

Doutor Mário Marques da Silva, Professor Associado (Universidade Autónoma de Lisboa)

Doutor Alberto Carneiro, Professor Associado (Universidade Autónoma de Lisboa)

Doutor Nuno Garcia dos Santos, Professor Auxiliar (Universidade da Beira Interior)

Lisboa, Maio 2014

Agradecimentos

Gostaria de começar por agradecer ao Professor Alberto Carneiro, pelo incentivo sistemático e amizade demonstrado ao longo deste trabalho e de toda a nossa convivência.

Também agradeço a todos os meus colegas, especialmente das disciplinas com quem tenho co-leccionado, que me receberam e ensinaram muito do que sei hoje. Espero poder contribuir ainda mais com aquilo que de novo aprendi.

Devido ao seu elevado valor curricular, agradeço a oportunidade que a Universidade Autónoma de Lisboa me deu em participar neste curso de Mestrado em Engenharia e Tecnologias Informáticas e a todo o grupo de docentes e convidados que partilharam connosco os seus conhecimentos e trabalhos, bem como aos meus colegas de curso, cuja experiência profissional também contribuiu para uma nova abordagem de diferentes temas.

Por fim, sublinho a valorização que este Mestrado me proporcionou, permitindo expandir os meus conhecimento na área da segurança em sistemas de armazenamento e podendo melhor contribuir para uma escola de tecnologias de informação, orientada para a excelência.

Resumo

Os sistemas de armazenamento de dados em *grid* tipicamente encontram-se fisicamente distribuídos e partilhados por vários utilizadores, o que pressupõe que apenas os utilizadores autorizados devam ter acesso à informação disponibilizada nesses sistemas, bem como garantir nesses mesmos sistemas a confidencialidade e a integridade de informação e dos dados.

Irá ser proposta neste trabalho a implementação de um mecanismo de segurança para *grids* de armazenamento, através de um algoritmo de criptografia simples que combina princípios de criptografia assimétrica (RSA) com a criptografia simétrica (AES, *Advanced Encryption Standard*). A ideia é cifrar a informação de cada utilizador armazenada no sistema, de modo a que os dados tenham um nível de protecção adicional, evitando assim que a solução de armazenamento fique comprometida caso seja exposto ou acedido um dos conteúdos, por parte de um utilizador que não esteja devidamente credenciado e autorizado para o efeito. Além disso é uma forma eficaz de garantir a confidencialidade dos dados mesmo que desagregando o cluster distribuído onde essa informação se encontra, que de outra forma se poderia tornar mais volátil, uma vez que a informação a nível do bloco deixava de ter a protecção de segurança do sistema operativo que a gere.

Deste modo consegue obter-se um sistema de ficheiros em *grid* com capacidade de encriptação, o que vai conferir maior segurança aos utilizadores e organizações que possam vir a aderir a esta tecnologia.

Abstract

Storage systems in data *grid* typically are physically distributed and shared by multiple users, which implies that only authorized users should have access to information available on these systems as well as in those systems ensure the confidentiality and integrity of information and data.

In this work, it will be proposed the implementation of a security mechanism for *grid* storage through a simple encryption algorithm that combines principles of asymmetric encryption (RSA, Rivest-Shamir-Adleman) with the symmetric cryptography (AES, Advanced Encryption Standard). The main idea is to encrypt the information stored for each user in the system, so that the data have an additional level of protection, thus preventing the storage solution is compromised if one of the exposed or accessed content from a user who not properly accredited and authorized for that purpose. Moreover it is an effective way to ensure the confidentiality of the data the disaggregation of the distributed cluster where the information is , that otherwise could become more volatile, since the information at block level no longer had the protection of security by the operating system.

Thus manages to obtain a file system on the *grid* with encryption capability, which will provide greater certainty for users and organizations that may join this technology.

Índice

Agradecimentos	i
Resumo.....	ii
Abstract	iii
Índice de tabelas	vi
Índice de figuras	vii
Glossário.....	viii
Prefácio	1
Parte I	5
Capítulo 1 – Introdução.....	6
Capítulo 2 – Revisão da literatura e métodos usados.....	9
2.1 - Conceitos.....	9
2.2 – Metodologia – <i>Design Science Research</i>	14
2.2.1 - Modelo <i>Design Science</i>	14
2.2.2. Diretrizes de Aplicação da Abordagem <i>Design Science</i>	15
2.2.3 – Aplicação da metodologia nesta investigação	18
Capítulo 3 – Mecanismo Criptográfico de Segurança de Armazenamento de Dados	21
Capítulo 4 – Implementação do Sistema de Segurança de Armazenamento em Grid	26
Capítulo 5 – Avaliação	35
Capítulo 6 – Conclusão	36
Parte II	37
Capítulo 7 – Curriculum.....	38
7.1. Dados Pessoais	38
7.2. Habilitações Literárias	38
7.2.1. Formação Universitária	38
7.2.2. Formação Complementar	39
7.3. Experiência Pedagógica.....	46
7.3.1. Área Educacional.....	46
7.3.2. Área de Informática.....	46
7.4. Experiência Profissional	51
7.5. Publicações.....	55
7.5.1. Capítulos de livros	55
7.5.2. Publicações de trabalhos completos em conferências	56
7.5.3. Artigos em Jornais	57

7.5.4. Participações em seminários como speaker	57
7.5.5. Submissão Papers, publicações e projetos de pesquisa	57
7.6. Informações Complementares	58
7.6.1. Conhecimentos de Francês e Inglês, escrito e falado.	58
7.6.2. Actividades Associativas:.....	58
7.6.3. Actividades Desportivas:	58
7.6.4. Outras Informações:.....	59
Referências bibliográficas	60

Índice de tabelas

Tabela 1 - Métodos de avaliação utilizáveis em Design Science.	16
Tabela 2 - Nível do Quadro Europeu Comum de Referência (CECR)	58

Índice de figuras

Figura 1- <i>SRM</i> como <i>infra-estrutura da grid</i>	11
Figura 2 - Arquitectura lógica de segurança.....	22
Figura 3 - Fase de Inicialização.....	23
Figura 4 - Algoritmos de primitivas	24
Figura 5 - Algoritmo e terminação	25
Figura 6 - a) Arquitectura e b) Sistema de Ficheiros.....	26
Figura 7 - Biblioteca de inicialização (a) primeira, (b) utilizações seguintes	28
Figura 8 - Primitivas de Dados I/O : (a) __leitura, (b) __escrita, (c) __<op>.....	30
Figura 9 - Implementação da Primitiva de Finalização	31
Figura 10 - Desempenho das operações	32

Glossário

AFS:	Andrew File System
API:	Application Programming Interface
BDII:	Berkeley Database Information Index
CASTOR	CERN Advanced STORage manager
CE:	Computing Element
CERN:	European Laboratory for Particle Physics
ClassAd:	Classified advertisement (Condor)
CLI:	Command Line Interface
CNAF:	INFN's National Center for Telematics and Informatics
dcap:	dCache Access Protocol
DIT:	Directory Information Tree (LDAP)
DLI:	Data Location Interface
DN:	Distinguished Name
EDG:	European DataGrid
EDT:	European DataTAG
EGEE:	Enabling Grids for E-science
ESM:	Experiment Software Manager
FCR:	Freedom of Choice for Resources
FNAL:	Fermi National Accelerator Laboratory
FTS:	File Transfer Service
GFAL:	Grid File Access Library
GG:	Grid Gate (aka gatekeeper)
GGF:	Global Grid Forum (now called OGF)
GGUS:	Global Grid User Support

GIIS:	Grid Index Information Server
GLUE:	Grid Laboratory for a Uniform Environment
GMA:	Grid Monitoring Architecture
GOC:	Grid Operations Centre
GRAM:	Grid Resource Allocation Manager
GRIS:	Grid Resource Information Service
GSI:	Grid Security Infrastructure
gsidcap:	GSI-enabled version of the dCache Access Protocol
gsirfio:	GSI-enabled version of the Remote File Input/Output protocol
GUI:	Graphical User Interface
GUID:	Grid Unique ID
HSM:	Hierarchical Storage Manager
ID:	Identifier
INFN:	Istituto Nazionale di Fisica Nucleare
IS:	Information Service
JDL:	Job Description Language
kdcap:	Kerberos-enabled version of the dCache Access Protocol
LAN:	Local Area Network
LB:	Logging and Bookkeeping Service
LDAP:	Lightweight Directory Access Protocol
LFC:	LCG File Catalogue
LFN:	Logical File Name
LHC:	Large Hadron Collider
LCG:	LHC Computing Grid
LRC:	Local Replica Catalogue
LRMS:	Local Resource Management System

LSF:	Load Sharing Facility
MDS:	Monitoring and Discovery Service
MPI:	Message Passing Interface
MSS:	Mass Storage System
NS:	Network Server
OGF:	Open Grid Forum (formerly called GGF)
OS:	Operating System
PBS:	Portable Batch System
PFN:	Physical File name
PID:	Process IDentifier
POOL:	Pool of Persistent Objects for LHC
PPS:	Pre-Production Service
RAL:	Rutherford Appleton Laboratory
RB:	Resource Broker
RFIO:	Remote File Input/Output
R-GMA:	Relational Grid Monitoring Architecture
RLI:	Replica Location Index
RLS:	Replica Location Service
RM:	Replica Manager
RMC:	Replica Metadata Catalogue
RMS:	Replica Management System
ROC:	Regional Operations Centre
ROS:	Replica Optimization Service
SAM:	Service Availability Monitoring
SASL:	Simple Authorization & Security Layer (LDAP)
SE:	Storage Element

SFN:	Site File Name
SMP:	Symmetric Multi Processor
SN:	Subject Name
SRM:	Storage Resource Manager
SURL:	Storage URL
TURL:	Transport URL
UI:	User Interface
URI:	Uniform Resource Identifier
URL:	Uniform Resource Locator
UUID:	Universal Unique ID
VDT:	Virtual Data Toolkit
VO:	Virtual Organization
WLCG:	Worldwide LHC Computing Grid
WMS:	Workload Management System
WN:	Worker Node
WPn:	Work Package #n

Prefácio

O sector financeiro sempre foi uma área bastante atractiva de uma perspectiva de infra-estrutura de TI's. O envolvimento do autor deste trabalho na área de sistemas de armazenamento era bastante grande uma vez que era a sua área de trabalho e também onde tinha tido já oportunidade de estar envolvido em actividades de investigação. Juntando estes dois argumentos e conhecendo a realidade do sector financeiro em matéria de sistemas de armazenamento, teve a possibilidade de reunir diversos materiais para desenvolver uma investigação mais formal. É nesse contexto e como passo inicial que nasce o trabalho de investigação que agora está a ser apresentado.

Depois de elaboradas as tarefas e agenda do projecto de investigação começámos por fazer um levantamento da realidade dos sistemas de informação no sector financeiro. O estudo da amostra foi sobretudo conduzido nos principais bancos de referência em Portugal. Foram feitas entrevistas exploratórias não estruturadas para perceber quais as tarefas diárias dos técnicos de sistemas de armazenamento e tentar isolar problemas que pudessem ser endereçados no âmbito desta investigação. A questão mais apontada por todos os entrevistados respeitava ao crescimento exponencial de dados, quer por via das operações regulares quer por via de questões de *compliance* com directrizes de regulamentação, que acarretam grandes problemas a nível da gestão diária dos sistemas de armazenamento, o que foi reconhecido através de entrevistas feitas a especialistas da área estratégica de TI's dos principais bancos portugueses.

Na pesquisa que aqui se iniciou, fomos tentar saber como outras organizações, que já de alguma forma se debatiam com este problema, embora por outros motivos, teriam endereçado a questão. Foi no mundo da física de altas energias, que lida com petabytes de informação (Sistemas Massivos de Armazenamento de Dados – MSS), que encontrámos alguns paradigmas inspiradores para lidar com este problema. Com efeito no CERN (e também noutros centros, nomeadamente RAL, Fermi Lab e Berkeley) é utilizada uma implementação do *middleware* SRM (*Storage Resource Manager*), que é na prática uma concepção de *grid* de armazenamento. A nossa pesquisa incidiu nos trabalhos de investigação que estavam a decorrer na área e no tipo de esforço seria necessário para trazer o SRM para o mundo do sector financeiro. Na data de início desta investigação não se registavam ainda aplicações de SRM ao sector financeiro, existindo apenas uma aplicação a sistemas actuários de análise de risco. Na altura pareceu ser ideia

válida e concluímos que poderia ser útil estender o SRM ao sector financeiro. Já com uma arquitectura SRM por nós desenvolvida, voltámos a contactar especialistas do sector financeiro, para avaliar qual seria a aceitação deste paradigma. Em todas as entrevistas, este teve uma aceitação entusiasmante, mas com a condição e garantias de capacidade de segurança evidente. Com esta incumbência, voltámos a fazer um levantamento do estado da arte do que está a ser feito em matéria de segurança no SRM, tendo verificado não haver investigação relevante. Quase todas as teses de doutoramento lidas referiam a questão de segurança como trabalho futuro. Assim ficou definido o focus de contribuição inovadora no trabalho de investigação que decorre - Segurança em *grids* de armazenamento do tipo SRM.

O *middleware* de *grid* tipicamente era desenvolvido no âmbito de pequenos projectos nos quais era possível implementar esquemas de autorizações simples, uma vez que a utilização típica das *grids* era mais para demonstrações tecnológicas do que para ambientes de produção. Nesse cenário estava envolvido apenas um pequeno número de utilizadores e as autorizações podiam ser geridas manualmente em cada recurso.

No Framework Globus era usado um mecanismo de autorizações simplificado - a *gridmap file*, uma lista simples, residente nos recursos, de utilizadores autorizados, expressos como Distinguished Names, associados com as credenciais locais correspondentes (ex. *usernames* em sistemas Unix).

Actualmente nas *grids*, o *middleware* tem tido contributos de diversas fontes. Neste cenário, com um grande número de utilizadores e de *sites*, um dos principais requisitos é o facto de a autorização de cada recurso ter de ser gerida por um procedimento automático, baseado numa ou mais políticas locais e em pontos centrais de autorização geridos manualmente.

Numa *grid*, os utilizadores estão normalmente organizados em entidades chamadas organizações virtuais (VO's). Uma VO é uma colecção de indivíduos e instituições que são definidas de acordo com um conjunto de regras e de partilha de recursos. Geralmente, as VO's partilham recursos e estabelecem acordos com instalações gerais chamadas *Resource Providers* (RP's) oferecendo recursos para a *grid* (ex. CPU, rede, armazenamento). Num ambiente potencialmente grande como uma *grid*, o problema da autorização de controlo de acesso para recursos é convenientemente simplificado baseado em VO's e RP's.

As infra-estruturas de segurança de *grid* baseiam-se em infra-estruturas de chaves públicas (PKI), para autenticação e autorização. O processo de autenticação é tipicamente deixado ao cuidado de entidades externas de confiança, como por exemplo Autoridades de Certificação; por outro lado, o processo de autorização é gerido, nos seus diferentes papéis, pelas VO's e pelos RP's. Especificamente, a informação geral que diz respeito à relação do utilizador com a sua VO (os atributos a que lhe são permitidos acesso) são geridos pela própria VO, ao passo que as RP's avaliam localmente esta informação tendo em consideração as políticas locais e as concordâncias com a VO, mapeando eventualmente as credenciais *grid* (certificados) para credenciais locais (credenciais Unix). Muitas áreas científicas utilizam grandes quantidades de armazenamento distribuído implementados por infra-estruturas em *grid*.

O sistema de armazenamento de informação em *grid* facilita a manipulação de grandes volumes de informação e disponibiliza funcionalidades de alto nível tais como distribuição e replicação de informação e acesso optimizado aos dados. Para construir um sistema de gestão de informação que se adapte a recursos de armazenamento de informação heterogéneos (discos, tapes ou silos), a comunidade *grid* adotou interfaces-standard para virtualizar os recursos subjacentes.

A primeira preocupação aquando do desenho do SRM, foi disponibilizar um acesso eficiente a grandes quantidades de informação e disponibilizar, entre outros serviços, o *pre-fetching* de ficheiros de informação gravados num armazenamento secundário, gestão do espaço de armazenamento e reserva de recursos de armazenamento. Todavia, não disponibiliza qualquer controlo de acesso ou protecção de informação, o que limita bastante o seu uso em aplicações que manipulem informação sensível.

Nesta dissertação, propomos uma resolução do problema da gestão de acesso a recursos na infra-estrutura *grid*. A abordagem inicial foi feita através de uma profunda análise de requisitos de gestão de informação no sector financeiro. Com base nesta análise, introduzimos um conjunto de serviços para dotarem a estrutura SRM de segurança no acesso, edição e armazenamento de informação. Estes serviços permitem um controlo de acesso à informação e aos metadados usando credenciais *grid standard* e disponibilizam em tempo real capacidades de encriptação / desencriptação. É também proposta uma arquitectura de *software* para implementar um serviço de armazenamento de dados para o sector financeiro. A nossa proposta disponibiliza serviços *grid* à

organização e seus utilizadores, que de alguma forma estão relacionadas entre si e tendo em conta as restrições relacionadas com a especificidade do sector.

As questões relacionadas com segurança e privacidade são habitualmente investigadas apenas no contexto da organização com o intuito de garantir o controlo de acesso dos utilizadores aos recursos. Para proteger os recursos de acesso não autorizado, são definidas políticas de segurança estaticamente dentro do limite lógico e físico da organização e normalmente geridas centralmente.

Este documento não aplica as regras do acordo ortográfico.

Parte I

Capítulo 1 – Introdução

As Tecnologias da Informação (TI) trouxeram uma nova tendência ao paradigma da computação que é a Computação em Redes Distribuídas. Entre elas, a *grid* é uma das mais difundidas. O seu sucesso deve-se ao facto de organizar e disponibilizar grandes quantidades de recursos computacionais e de armazenamento para atribuição e processamento de dados, conforme exigido pelos utilizadores e conforme a necessidade de fluxos de computação. A gestão de tais recursos é transparente para o utilizador que só tem de especificar os seus requisitos em termos de recursos. O sistema de gestão da *grid* determina automaticamente onde é executado o processo e que recursos devem ser disponibilizados (Arvidson, 2003, Foster and Kesselman, 1999). A partilha de dados distribuídos em ambientes multi-utilizador desencadeia problemas de segurança, nomeadamente problemas de confidencialidade e de integridade de dados. Os *Middleware Grid* geralmente oferecem capacidade de gestão de recursos, garantindo segurança no acesso aos serviços e na comunicação de dados. Este serviço, devido a acessos indevidos ao nível do sistema, facilita muitas vezes a vulnerabilidade dos dados. Por outras palavras, o facto de os dados serem distribuídos e armazenados em máquinas remotas e distribuídas, onde os respectivos administradores acedem directamente, constitui o principal risco para a segurança dos dados em ambiente de *grid*. Problemas de segurança, tais como abuso/ataque a informação privilegiada, roubos de identidade e / ou apropriação de contas, muitas vezes não são adequadamente cobertos no contexto da *grid*. Portanto, torna-se obrigatória a introdução de um mecanismo de protecção adequada de dados, que impeça a compreensão dos dados por parte de utilizadores não autorizados, mesmo que sejam administradores locais do sistema.

O problema do acesso a um armazenamento seguro é referenciado, sobretudo na literatura, como definição de direitos de acesso (Junrang *et al.*, 2004), em especial, abordando problemas de partilha de dados, ao passo que a codificação dos dados é exigida ao utilizador, visto não ter sido ainda definido nenhum mecanismo de acesso automático para o espaço de armazenamento que o torne seguro e transparente.

Scardaci e Scuderi (2007) propuseram uma técnica para proteger os dados distribuídos em ambiente *grid* SRM (Sim *et al.*, 2005), baseado em criptografia simétrica (*Advanced Encryption Standard, AES*). A chave de segurança é confiada a um único servidor de armazenamento de chaves que a armazena e todos os pedidos de acesso aos

dados implicam a notificação deste servidor para decifrar os dados. Este algoritmo implementa uma política de segurança espacial: a segurança é conseguida escondendo e protegendo fisicamente o servidor de armazenamento da chave; o acesso ao armazenamento de chaves é fisicamente restrito e monitorizado, para o proteger de utilizadores maliciosos, ataques externos também e abusos a partir de dentro do sistema. Seitz, Pierson e Brunie (Seitz *et al.*, 2003) estudaram o problema de acesso a dados e propuseram uma solução baseada em chaves simétricas. A fim de evitar acessos não autorizados à chave simétrica, os autores propõem subdividi-la por diferentes servidores. Uma técnica semelhante foi especificada por Shamir (1979), usado no sistema de autenticação Perroquet (Blanchet *et al.*, 2006) para modificar o *middleware* PARROT (Thain and Livny, 2005) adicionando um gestor de ficheiros encriptados (*encrypted file manager*). A principal contribuição desse trabalho é que, aplicando o algoritmo proposto, a chave simétrica (AES), dividida em N partes, pode ser recomposta se, e somente se, todas as N partes estão disponíveis. O sistema HYDRA (2010) implementa um serviço de partilha de dados em ambiente médico utilizando por ao *middleware* SRM, protegendo dados através do uso da criptografia simétrica e divide as chaves por três servidores de chaves (Montagnat *et al.*, 2006).

Todas as propostas acima mencionadas são baseadas em criptografia simétrica. A maioria delas implementa algoritmos de divisão de chaves. A ideia subjacente da abordagem “chave dividida” é que, pelo menos, um subconjunto dos sistemas (servidores de chave) sobre os quais as chaves são distribuídas será confiável. No entanto, esta abordagem é fraca sob três pontos de vista:

1. A segurança - as listas de servidores com partes da chave devem ser adequadamente protegidas. No entanto os administradores do sistema podem sempre aceder às chaves, sendo difícil atingir a confiança nos nós remotos e distribuídos;
2. A fiabilidade / disponibilidade - se um dos servidores armazenar uma parte da chave e não estiver disponível, os dados não podem ser acedidos;
3. A performance - há uma sobrecarga inicial para reconstruir uma chave, dependendo do número de partes em que a chave é dividida. Uma solução para melhorar a confiabilidade / disponibilidade é replicar os servidores chave, mas isto contrasta com os desafios de segurança.

O objectivo do nosso trabalho é fornecer um mecanismo capaz de armazenar dados em ambiente *grid* de uma forma segura. Para fazer isso, propomos combinar a criptografia simétrica e a assimétrica. Portanto, a principal contribuição do trabalho é a especificação de uma técnica simples e eficaz para armazenamento seguro de dados num ambiente *grid* que conjuga o objectivo da alta segurança com problemas de desempenho, como também tem sido sugerido (Zeng *et al.*, 2009). A técnica da arquitectura de segurança que propomos foi implementada no *middleware* SRM, a fim de demonstrar a viabilidade da abordagem pelos resultados obtidos, através da avaliação de desempenho desta arquitectura de segurança. Outras contribuições interessantes desta arquitectura de segurança são a organização dos dados numa *grid* num sistema de ficheiros e a protecção de dados e ficheiros na própria estrutura do sistema de ficheiros.

Este trabalho encontra-se organizado em duas partes. Uma primeira parte com uma implementação tecnológica e uma segunda parte com uma descrição sumária do *curriculum vitae* do investigador. Desta forma na primeira parte e após uma breve introdução de conceitos básicos na secção seguinte, descrevemos o algoritmo usado bem como a sua implementação no *middleware* SRM. Em seguida, mostram-se os resultados obtidos através da avaliação da nossa implementação. A discussão sobre vantagens e desvantagens desta técnica é desenvolvida no ponto seguinte e a secção final propõe algumas considerações e indica possíveis trabalhos no futuro.

Na segunda parte deste trabalho encontra-se o *curriculum vitae* do investigador onde se pode ter uma noção da evolução do percurso empresarial que de alguma forma mapeia o amadurecimento de conhecimentos tecnológicos e experiência tecnológicas e científicas por mais de duas décadas.

Capítulo 2 – Revisão da literatura e métodos usados

O conceito de *grid* compreende uma arquitectura de protocolos distribuída, que permite o uso de um conjunto de recursos de computação e de armazenamento dispersos geograficamente como se de um único sistema se tratasse (Foster and Kesselman, 1998, Foster *et al.*, 2002). A colecção de recursos de *software*, serviços, API, primitivas, comandos, ferramentas, protocolos e interfaces para o gerir ambientes de *Grid Computing* são geralmente agrupados num único *middleware*. Diferentes *middlewares de grid* têm sido implementados, quer a título comercial quer a título de *freeware*. Entre eles, o Globus (Globus, 2013) é um dos mais difundidos. Outros igualmente conhecidos são o Condor (Thain and Livny, 2005), o BOINC (Berkeley Open Infrastructure for Network Computing) (Anderson, 2004) e o SRM (Sim and Berkeley, 2008), este último desenvolvido a partir de protocolos e serviços do Globus.

2.1 - Conceitos

Num sentido mais amplo, o termo *grid* significa uma infra-estrutura computacional específica destinada a agregar um conjunto de recursos. As *grids* podem ser classificadas por recurso, escala e serviços. De acordo com a taxonomia de classificação de *grids*, considerando o tipo de recursos que ela partilha, podem ser classificada em quatro tipos:

1. *Grid* de Computação: Se os principais recursos partilhados são a CPU;
2. *Grid* de Dados: Quando o objectivo principal é partilhar recursos de dados, tais como os resultados de experiências, entre os utilizadores;
3. *Grid* de Armazenamento: criada para oferecer aos utilizadores o acesso a uma enorme quantidade de espaço de armazenamento;
4. *Grid* de Equipamento: também pode ser configurado para partilhar o acesso a recursos físicos, nomeadamente telescópios astronómicos e satélites.

No entanto, algumas *grids* podem ser inseridas em mais do que uma destas categorias. As *grids* de computação também podem ser classificadas pela forma como geograficamente distribuem os seus recursos:

1. *Grids* à escala da Internet podem incluir qualquer pessoa com acesso à Internet;
2. *Grids* de Organização Virtual (VO) contêm várias entidades académicas ou corporativas;

3. *Grids* locais que estão contidas dentro de uma organização.

Em cada um destes casos, as empresas têm acesso aos seus próprios *clusters* para realizar tarefas de processamento. Da perspectiva do utilizador, o mais importante sobre a *grid* são os serviços que disponibiliza. A *grid* de dados permite o acesso a recursos específicos de dados. Neste momento, os tipos de serviço mais comuns oferecidos por sistemas *grid* são: processamento gráfico, simulações científicas e aplicações web.

Para utilizar todo o seu potencial, a *grid* exige soluções para questões fundamentais como autenticação, autorização, descoberta de recursos, acesso a recursos e principalmente, a incompatibilidade de recursos e políticas para a sua gestão. Uma grande variedade de projectos e produtos oferecem serviços destinados a abordar estas questões. O *Globus Toolkit* (Foster and Kesselman, 1998) é um *software open-source* que simplifica a colaboração dinâmica entre organizações virtuais (VO) multi-institucionais. As ferramentas incluem *software* de serviços e bibliotecas para monitorização de recursos, gestão e descoberta de recursos e gestão de segurança, infra-estrutura da informação, gestão de dados e ficheiros, comunicação, deteção de falhas e portabilidade. O *Globus* inclui um conjunto de componentes que podem ser usados de forma independente ou em conjunto para desenvolver aplicações. O *Globus Toolkit* foi concebido para remover os obstáculos que impeçam uma correcta colaboração. Os seus principais serviços, interfaces e protocolos permitem aos utilizadores aceder a recursos remotos como se estivessem localizados na sua própria máquina, enquanto preservam o controlo local sobre quem e quando pode usar os recursos.

O *Globus Toolkit* tem evoluído através de uma estratégia de código aberto semelhante ao do sistema operativo Linux, ao contrário de outros sistemas proprietários de *software* de partilha de recursos. Outro *middleware* para gestão do ambiente *grid* é o *SRM* (Sim and Berkeley, 2008). Uma vez que este trabalho foi implementado a partir do *middleware SRM*, centramo-nos principalmente neste *middleware*.

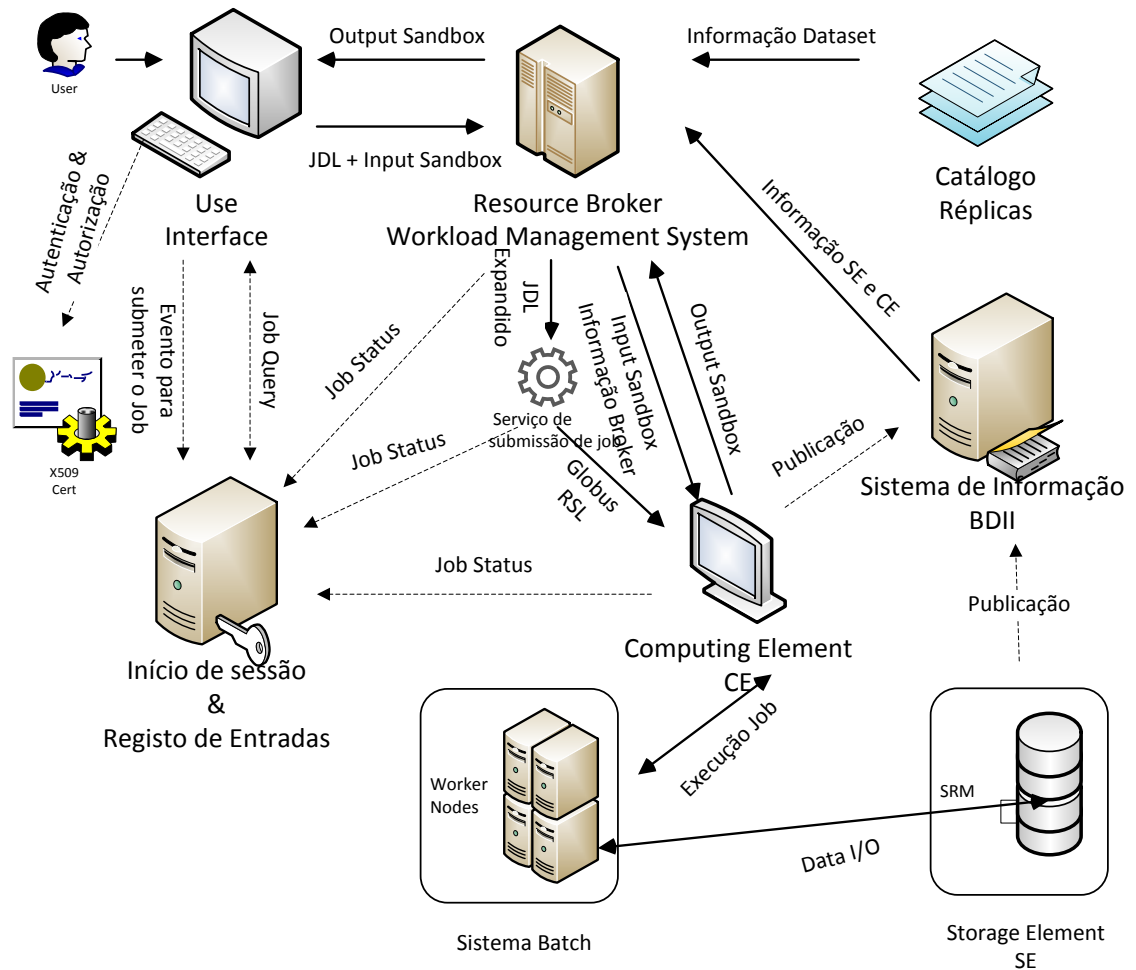


Figura 1- SRM como infra-estrutura grid

O Middleware SRM

O SRM – *Storage Resource Manager* (Sim and Berkeley, 2008) é um *middleware* para a gestão de infra-estruturas de *Grid Computing*, desenvolvido em *Berkeley* e com contributos de vários outros laboratórios de investigação, nomeadamente o *CERN*, com o objectivo de implementar uma computação altamente distribuída, poderosa e flexível e, no caso do *CERN*, uma plataforma de armazenamento de dados para armazenamento de dados produzidos pelo *Large Hadron Collider (LHC)* (2010) (ver Figura 1).

- Os Serviços do SRM abrangem a segurança, monitorização, gestão de tarefas e de dados, desenvolvido seguindo uma *Arquitectura Orientada a Serviços (SOA)*:
- O serviço de gestão de tarefas é hierarquicamente implementado e dividido entre os elementos de computação (CE) e Nós de Trabalho (WN). Os elementos de

computação recolhem as tarefas enviadas à infra-estrutura *Grid SRM* em filas específicas e depois envia esses serviços para que os Nós de Trabalho (WN) as processem.

- Segurança. A autenticação tem por base a infra-estrutura *Globus Security Infrastructure* (GSI, 2010), que usa certificados X.509 (Perlman *et al.*, 2004) para implementar mecanismos de delegação de credenciais na autenticação do utilizador (*Single Sign-On*). Os recursos da *grid* são agrupados em organizações virtuais (VO's) e há um sistema de gestão das organizações virtuais (VOMS) (Alfieri *et al.*, 2003) para garantir a política de segurança num ambiente distribuído que abranja recursos de *hardware* diferentes e em diferentes locais.
- Dados: A informação está localizada sobre os recursos de armazenamento chamado elementos de armazenamento (*Storage Element* (SE)), que gere sistemas de discos e *hardware* necessário para armazenar dados.

Todos os recursos na *grid* são geridos pelo *Resource Broker* (RB) de forma centralizada: todo o trabalho de execução de pedidos é dirigido para ele, que verifica os recursos necessários para o processamento de tais serviços (descoberta de recursos), localiza-os e transfere o contexto do trabalho para o recurso correspondente. O ponto de acesso dos serviços *SRM* é a interface de utilizador (*User Interface*) (UI), que consiste numa colecção de utilitários para interagir com a infra-estrutura de *Grid SRM*.

Grid File Access Library (GFAL)

As interações com a *grid* de armazenamento requerem o uso de vários componentes de *software*: o serviço de réplica de catálogo para localizar réplicas válidas dos ficheiros, o gestor de recursos de armazenamento (SRM) que assegura que os ficheiros existem no disco (ou obtidos de meios de armazenamento) ou a reserva de espaço em disco para novos ficheiros (que poderão ser migrados posteriormente para meios de armazenamento). O SRM garante também o acesso a ficheiros no sistema de armazenamento no nó de trabalho.

A biblioteca *Grid File Access Library* (GFAL) esconde estas interações implementando uma interface *Posix* para as operações de I/O. Os protocolos suportados são: *file* para o suporte de acesso local tipo “*nfs*”, *dcap*, *gsidcap* e *kdcap* (protocolo de acesso *dCache*) e *rfio* (protocolo de acesso CASTOR). As funções implementadas nas biblioteca GFALsão chamadas através da adição do prefixo “*gfal*” aos nomes das funções

Posix, como por exemplo *gfal_open*, *gfal_read*, *gfal_close*. Os argumentos, assim como os valores devolvidos são idênticos aos das funções originais *Posix*. O GFAL aceita as seguintes convenções de nomes: *Logical File Name* (LFN), *Grid Unique Identifier* (GUID), *Storage File Replica* (SURL) ou *Transport File Name* (TURL).

Identificador Único Universal

O Identificador Único Universal (UUID) é um mecanismo distribuído de identificação especificado por Leach, Mealling & Salz (Leach *et al.*, 2005), que garante identificadores únicos sem um processo de registo ou coordenação centralizado. É utilizado em engenharia de *software*. Um UUID é um número de 16 bytes (128 bits). Na sua forma canónica consiste em 32 caracteres hexadecimais, agrupados em cinco grupos, separados por hífen, somando desta forma um total de 36 caracteres.

No contexto da *grid*, os UUID são por vezes identificados com um acrónimo de GUID que significa Identificador Único Global ou Identificador Único da *Grid*.

Réplica

No EDG *Manager* e no *Globus Replica Location Service*, o GUID é essencial para estabelecer a ligação entre o *Replica Metadata Catalog* (que contem o nome do ficheiro lógico) e o *Local Replica Catalog* (que contem o URL do armazenamento). Nos catálogos LCGFile mais recentes – *LCGFile Catalog* e *FiReMan*, não é necessário que o utilizador conheça o GUID visto que as tabelas que contêm o nome lógico do ficheiro e o URL do armazenamento estão contidos na mesma base de dados.

Criptografia

Em relação à segurança dos dados, escolheu-se implementar nesta arquitectura uma técnica criptográfica. Existem dois tipos de sistemas criptográficos: simétrico (também conhecido por convencional ou chave secreta) e assimétrico (chave pública). A cifragem simétrica requer que o emissor e receptor tenham a mesma chave. Esta chave é usada pelo emissor para encriptar os dados e pelo receptor para efectuar a descriptação dos dados. O algoritmo simétrico mais usado é o Advanced Encryption Standard (AES) (NIST, 2001) também conhecido por *Rijndael*. Com as cifras assimétricas cada utilizador tem um conjunto de chaves: uma chave pública e uma chave privada. As mensagens encriptadas com uma chave apenas podem ser descriptadas pela outra chave. A chave pública pode ser publicada – de domínio público, enquanto a chave privada é mantida em segredo. Um dos algoritmos assimétricos mais interessantes é o RSA (Rivest *et al.*, 1978).

As operações de cifra assimétricas são bastante mais lentas e o tamanho das suas chaves tem de ser superior às usadas nas operações de cifra simétrica. De momento, para quebrar quer os algoritmos AES quer o RSA, apenas a força bruta é eficaz e é necessário grande capacidade de processamento e bastante tempo para se obter a chave, especialmente para o caso do RSA. Uma técnica que combina e sintetiza a alta segurança da criptografia assimétrica com a eficiência da criptografia simétrica é a PGP (*Pretty Good Privacy*) (Garfinkel, 1994). Neste algoritmo os dados são cifrados com uma chave simétrica e posteriormente é aplicado um algoritmo assimétrico. Um algoritmo semelhante ao PGP foi desenvolvido pelo GNU no projecto de código aberto GPG (GNU Privacy Guard, 2010).

2.2 – Metodologia – *Design Science Research*

A metodologia DSR (*Design Science Research*) foi a metodologia usada nesta investigação e seguida ao longo deste trabalho. A metodologia DSR tem como objectivo principal desenvolver conhecimento que possa ser utilizado por profissionais na sua área de actuação a fim de solucionar problemas, o termo *Design Science* destaca a orientação do conhecimento actual ao *Design* (Soluções de problemas do mundo real) e as ferramentas que serão utilizadas para acções adequadas de domínio de profissionais (Sordi *et al.*, 2011). Vale a pena realçar que existe uma diferença entre a abordagem *Design Science*, com as práticas de *design* de produtos, que envolve protótipos com a função de testes, averiguando então a aceitação de inovações tecnológicas.

2.2.1 - Modelo *Design Science*

A origem do conceito de *Design Science* é atribuída directamente a Buckminster Fuller (Fuller and Mchale, 1963) e a sua popularização a Herbert Simon, através do seu livro *The Sciences of the Artificial* (Simon, 1967), que estabelece a diferença entre as “Ciências Naturais” que estão ligadas ao estudo de como são constituídos e o funcionamento dos objetos naturais e sociais e as “Ciências do Artificial” que se ocupam da concepção e construção de artefactos, podendo ser numa definição ampla, estruturas, modelos, métodos, instanciações ou mesmo novas propriedades de recursos técnicos, sociais, ou informacionais (Hevner *et al.*, 2004 2004).

2.2.2. Directrizes de Aplicação da Abordagem *Design Science*

Alguns autores criaram um conjunto de sete directrizes, tornando como referência para pesquisadores, revisores, editores e leitores no que favorece a compreensão e avaliação do método de pesquisa *Design Science* (Hevner *et al.*, 2004). Estas directrizes devem ser cuidadosamente observadas em toda e qualquer pesquisa que usa a abordagem *Design Science*.

Directriz 1: *Design Science* utilizando um artefacto como objeto de estudo.

De acordo com Simon (Simon, 1967), artefacto é tudo o que não é considerado natural, algo construído pelo homem. Os princípios da metodologia *Design Science* têm as suas raízes na engenharia das coisas artificiais e os Sistemas de Informação são um exemplo do que caracterizamos de Sistemas Artificiais. Sabemos que os Sistemas de Informação são criados dentro de uma organização com o objectivo de incrementar a eficiência da mesma. Estes sistemas não seguem as leis naturais ou as teorias comportamentais, ao contrário disso, a criação deles é feita confiando num núcleo de teorias que são aplicadas, testadas, modificadas e expandidas por meio da experiência, criatividade, intuição e capacidade de resolver problemas do investigador que está presente (Walls *et al.*, 1992). No campo de Tecnologia de Informação, os principais artefactos seriam construtores, modelos, métodos e geradores de instâncias, estes em *Design Science* são exemplos concretos, como modelos ou protótipos.

Directriz 2: O problema como foco principal.

A equipa de Hevner confirma que o problema precisa de ser a principal motivação, deve haver maior interesse e sua solução ser útil para os utilizadores (Hevner *et al.*, 2004). A *Design Science* é voltada para as soluções de base tecnológica e designada a importante problemas empresariais. O problema pode ser definido como a diferença entre um objetivo-meta e o estado corrente de um sistema (Simon, 1996). A resolução do problema, feita desta forma, consiste em desenvolver acções para reduzir ou eliminar as condições necessárias de actuação, impostas pelos grupos de poder; são eles, Clientes (qualidade, preço); Empregados (salários, empregos); Governo (impostos, condições de trabalho). Tais objectivos da organização criam problemas empresariais e oportunidades relacionadas com o custo e a renda. Sendo assim, os Sistemas de Informação passam a ser convocados para contribuir para tais objectivos.

Directriz 3: Avaliação

Segundo a equipa de Hevner, a utilidade, qualidade e eficácia da *Design Science* devem ser demonstradas rigorosamente, através de métodos precisos para a avaliação do resultado produzido (Hevner *et al.*, 2004). O componente crucial do processo de pesquisa é a avaliação. A avaliação do resultado da *Design Science* é frequentemente fundamentada nas exigências empresariais e ocorrem no contexto da utilidade, qualidade e beleza do artefacto produzido. Dentro da avaliação, existe também a integração entre o artefacto e a infra-estrutura técnica do ambiente do negócio. Segundo Johansson (Johansson, 2000), no ambiente da TI, os artefactos avaliados em termos de funcionalidade, perfeição, consistência, precisão, desempenho, confiabilidade, ajuste à organização e outros atributos de qualidade pertinentes. A avaliação de artefactos projetados é criada por meios de metodologias disponíveis na área científica. A escolha do método que será empregado é importante, pois ele deve ser o mais apropriado aos objectivos da avaliação.

Métodos de Avaliação	
Observação	Estudo de caso: estudo profundo do artefacto no ambiente da empresa Estudo de campo: monitorar o uso do artefacto em múltiplos projectos
Analítica	Análise estática: exame da estrutura do artefacto referente a qualidades estáticas (por exemplo: complexidade) Análise da arquitectura: estudo do ajuste do artefacto à arquitectura do SI Optimização: demonstração da optimização das propriedades do artefacto Análise dinâmica: estudo das qualidades dinâmicas do artefacto em uso (exemplo: performance)
Experimental	Experiência controlada: estudo do artefacto em ambiente controlado para análise das suas propriedades, como por exemplo: usabilidade Simulação: análise do artefacto com dados artificiais
Testes	Teste funcional (<i>Black Box</i>): execução do artefacto para descobrir falhas e identificar defeitos por meio de dispositivos específicos. Teste estrutural (<i>White Box</i>): teste de desempenho em relação a métricas na implementação do artefacto (por exemplo: teste de endereços)
Argumentação	Argumentação: uso de informação com base científica para construir um argumento convincente da utilidade do artefacto Cenários: construção detalhada de cenários em torno do artefacto para demonstrar sua utilidade

Tabela 1 - Métodos de avaliação utilizáveis em Design Science.

Fonte: Hevner et al. (2004)

Directriz 4: Contribuição da *Design Science* para a área de conhecimento do artefacto.

Em qualquer tipo de pesquisa, uma questão fundamental é saber quais são as contribuições inovadoras e interessantes para que a pesquisa proporciona? Segundo a mesma equipa de investigadores, a *Design Science* tem potencial para produzir três tipos de contribuições que são baseadas na inovação, generalidade e importância do artefacto projetado (Hevner *et al.*, 2004). Um ou mais destes tipos de contribuição devem ser considerados na pesquisa:

- Projeto do artefacto:
 - A contribuição da metodologia *Design Science* é a criação do artefacto. O artefacto deve ser a solução para o problema até então não solucionado. A pesquisa de Sistemas de Informação abrange a aplicação em ambiente apropriado. As metodologias para o desenvolvimento de sistemas, projetos de ferramentas e protótipos de sistemas são exemplos de artefactos.
- Ampliação dos fundamentos:
 - Os resultados da metodologia *Design Science* possibilitam fazer adições à base de conhecimento já existente. Os resultados podem ser a definição de construtores, métodos ou extensões de técnicas que melhorem as teorias, as estruturas, os instrumentos, conceitos, modelos, métodos e protótipos já existentes ou incrementem a base de conhecimentos referentes a técnicas de análise de dados, medidas, procedimentos e critérios de validação.
- Desenvolvimento de novas metodologias:
 - A criatividade de desenvolvimento e o uso de métodos de avaliação possibilitam a pesquisa *Design Science* e constituem-se em contribuição para expandir a base de conhecimento já existente. Em Sistemas de Informação, os artefactos precisam de representar o negócio, o ambiente em que a tecnologia é aplicada e os SI são modelos de negócios. Sendo assim, a pesquisa deve demonstrar uma clara contribuição para o ambiente empresarial, resolvendo um problema importante até então em aberto.

Directriz 5: Investigação Rigorosa

Uma pesquisa por meio de *Design Science* necessita de aplicação de métodos rigorosos na construção e na avaliação do projecto do artefacto. O rigor é avaliado pela adesão da pesquisa a uma colecção de dados apropriada e a análises técnicas corretas.

Directriz 6: Eficiência na utilização de recursos

Recursos disponíveis são empregados para alcançar os fins, satisfazendo as leis do ambiente pertinente ao problema. Uma pesquisa bem conduzida necessita de conhecimento no domínio de aplicação e no domínio de solução. Através das características específicas desta Directriz, a sua constatação em qualquer trabalho só pode ser feita se houver declaração específica dos autores que são referentes ao assunto.

Directriz 7: Comunicação de resultados

Os resultados obtidos da pesquisa *Design Science* são apresentados a diversas audiências com detalhes adequados a cada uma. Durante as apresentações são considerados detalhes específicos de acordo com o público-alvo.

2.2.3 – Aplicação da metodologia nesta investigação

O objectivo da implementação de um Sistema de Informação numa organização é melhorar a eficácia e a eficiência da mesma.

Seguimos o Framework DSR para compreensão, execução e avaliação da investigação que consta neste trabalho.

A ciência do *Design* procura criar inovações que definam ideias, práticas, capacidades técnicas e produtos através dos quais a análise, projecto, implementação e uso de sistemas de informação podem ser efectiva e eficientemente usados. Em DSR são criados e avaliados artefactos em TI para resolver problemas organizacionais reconhecidos. Um artefacto em TI é considerado como o objecto central das investigações em Sistemas de Informação. São definidos como construções, modelos, métodos e instanciações. As construções formam a linguagem na qual os problemas são definidos e comunicados. Os modelos usam construções para representar uma situação do mundo real, auxiliam a perceber o problema e a sua resolução e frequentemente representam a conexão entre o problema e as soluções permitindo a exploração dos efeitos das decisões do *design* e as mudanças no mundo real. Métodos definem processos, fornecem

directrizes sobre o modo como resolver os problemas. Instanciações mostram como construções, modelos e métodos podem ser implementados num sistema de trabalho. Mostram a viabilidade, permitindo uma correcta avaliação da adequação do artefacto à sua finalidade.

Evidências das directrizes de acordo com a metodologia DSR

Vamos evidenciar as directrizes da metodologia DSR nas acções levadas a cabo na investigação constante deste trabalho.

1. A primeira directriz assume que o artefacto, como objecto final da investigação, inclui construções, modelos, métodos e instanciações. De facto neste trabalho foi criada uma instanciação de um construtor, que permitiu criar uma experiência funcional onde foi gerado um artefacto de segurança, cuja implementação foi testada, gerando assim evidência do seu contributo utilitário no universo de utilização;
2. A segunda directriz indica-nos que a investigação deve tratar de um problema relevante. Efectivamente, questões relacionadas com sistemas de armazenamento são um dos problemas principais dos dias de hoje;
3. Como terceira directriz verificamos que é preciso avaliar o artefacto criado, esta avaliação pode ser feita de várias formas. A utilidade, qualidade e eficácia de um artefacto deve ser rigorosamente demonstrado através de métodos de avaliação bem executados. A avaliação foi feita pela utilidade e pelo tratamento de desempenho depois de implementado. A validação foi reconhecida por um painel de especialistas que nos manifestou grande aceitação na arquitectura apresentada bem com na solução de controlo de acesso desenvolvida e que foi o principal alvo desta investigação;
4. A quarta directriz estabelece que a investigação deve fazer alguma contribuição clara e verificável. O desenvolvimento de um sistema criptográfico enquadra-se nesta directriz e como foi verificado na fase de avaliação foi reconhecida uma contribuição clara para a problemática da escalabilidade vertical dos sistemas de armazenamento bem como dos seus mecanismos de segurança;
5. A quinta directriz refere-se ao rigor da investigação que deve ser comprovado pela utilização de fontes amplamente aceites. Foi feito uma rigorosa revisão bibliográfica e seguiram-se os procedimentos de boas práticas na investigação seguindo a metodologia que descrevemos, tais como a formulação de hipóteses,

experimentação, testes e verificações e por fim a abertura à generalização da utilização de um sistema criptográfico numa *grid* de armazenamento;

6. A sexta diretriz assume que o *design* é um processo de procura. As estratégias de procura buscam soluções boas e viáveis que possam ser implementadas num ambiente real. O processo de procura cria um ciclo de *design*. Simon descreve a natureza de um processo de *design* como um ciclo Gerar/Testar (Simon, 1996). O que de facto se verifica pois esta é uma fase embrionária de uma instanciação que poderá e de verá ser melhorada futuramente, tendo por base as recomendações de trabalhos futuros a seguir descritos;
7. Finalmente a sétima diretriz refere que a investigação deve ser comunicada de forma que seja compreendida por audiências técnicas e administrativas. Neste contexto foi feito um trabalho de simplificação do texto apresentado realçando o problema a resolver bem como a aplicabilidade do produto final da investigação.

Capítulo 3 – Mecanismo Criptográfico de Segurança de Armazenamento de Dados

O principal objectivo deste trabalho é implementar um sistema de segurança numa *grid* de armazenamento. As *grids* de armazenamento caracterizam-se pela capacidade de disponibilizar aos utilizadores grande capacidade de armazenamento. Neste contexto, a confidencialidade e a integridade dos dados têm de ser garantidas, de forma a prevenir ataques externos ou internos: a ideia é de que com excepção do utilizador/proprietário dos dados ninguém lhes possa aceder, incluindo os administradores de sistema.

A criptografia é uma excelente opção garantir a confidencialidade e segurança dos dados. Como descrito na introdução, até agora a forma adoptada para resolver este problema é a cifragem simétrica, devido ao seu desempenho face ao algoritmo assimétrico. A melhor abordagem é assim cifrar os dados usando o algoritmo simétrico (*DataKey*) e transferindo o problema de segurança para o tópico de ocultação eficaz desta chave (princípio de Kerckhoffs's) (Kerckhoffs, 1883).

Em relação à ocultação da chave, um algoritmo de divisão da chave é a solução que parcialmente atinge estas questões de segurança, como vimos na introdução. As vulnerabilidades tais como abusos internos, apropriação de contas e/ou furtos de identidade, não são adequadamente endereçados por esta solução, pois os administradores podem aceder aos componentes da chave. É assim necessária uma solução mais eficaz. Por esta razão, para ocultar a *DataKey*, propõe-se a sua encriptação pela chave pública do utilizador, explorando um algoritmo de cifragem assimétrico. Desta forma, apenas o utilizador/dono dos dados consegue ter acesso. Nesta secção, fornece-se uma descrição lógica da abordagem desta investigação. Nas subsecções subsequentes descreve-se o algoritmo de segurança, enquanto se fornece detalhes acerca da arquitectura que coloca em prática esse algoritmo.

Arquitectura Lógica de Segurança

A arquitectura investigada combina uma criptografia simétrica e assimétrica numa estrutura hierárquica, garantindo alta segurança. A arquitectura lógica desta abordagem é retratada na figura 2.

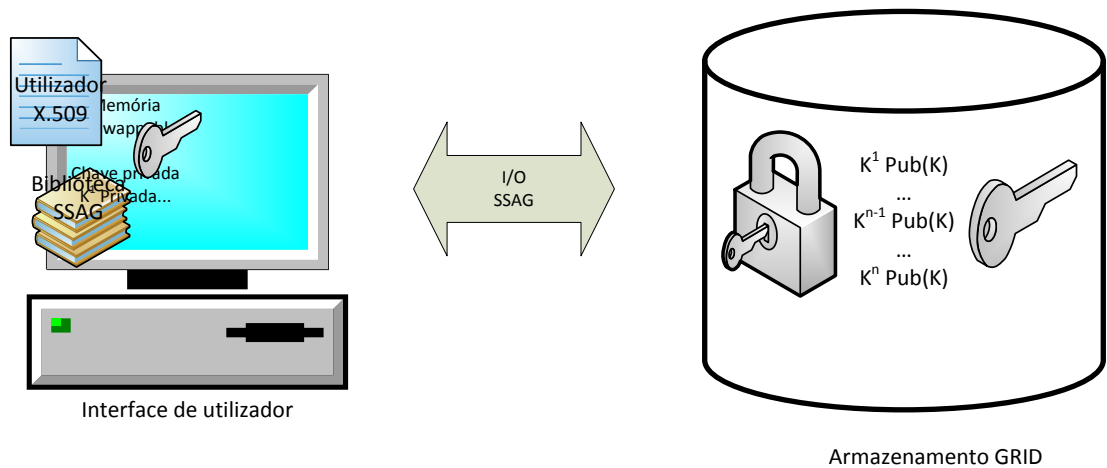


Figura 2 - Arquitectura lógica de segurança

Um utilizador autorizado, autenticado pelo seu certificado X509 e ambiente gráfico, contacta a *grid* de armazenamento onde os seus dados residem. Estes estão encriptados através do algoritmo simétrico, cuja chave simétrica *DataKey* (K) está também guardada na *grid* de armazenamento, por seu turno encriptada pela chave pública (K_{Pub}) do utilizador/dono, obtendo-se a chave encriptada *DataKey* $K_{Pub}(K)$. Desta forma, apenas o utilizador que tem a chave privada correspondente K_{Priv} pode desencriptar a chave simétrica e, portanto, os dados cifrados. A chave cifrada *DataKey* $K_{Pub}(K)$ é guardada em conjunto com os dados de forma a permitir ao utilizador/dono o acesso aos dados a partir de qualquer nó da infra-estrutura de *grid* de armazenamento. Um utilizador irá necessitar de um *Smartcard* que contem a chave privada. Para possibilitar a partilha de dados entre utilizadores, a *DataKey* K é guardada na *grid* de armazenamento e replicada em tantas cópias quantos os utilizadores autorizados a acederem aos dados. Desta forma, como se mostra na figura 2, uma cópia da *DataKey* encriptada pela chave pública do n -ésimo utilizador $K_n Pub(K_n Pub(K))$ tem de ser guardada na *grid* de armazenamento para permitir que esse utilizador aceda aos dados.

Note-se que, no algoritmo proposto, a descriptação é exclusivamente efectuada no nó do utilizador onde o correspondente certificado X509 se encontra. A chave simétrica desencriptada, os dados e toda a restante informação referente a estes, são mantidos numa zona de memória estática para evitar acessos indevidos. Desta forma, é obtido um nível mais elevado de segurança: dados e chaves são sempre encriptados quando estão longe do utilizador, quer em transferências de informação, quer em

armazenamento remoto; estão descriptados quando chegam ao nó do utilizador – sendo mantidos em áreas de memórias estáticas.

Algoritmo

Do ponto de visto algorítmico, a arquitectura lógica de segurança descrita anteriormente pode ser decomposta em dois passos: (i) a encriptação simétrica da *DataKey* K através da chave pública do utilizador K_{Pub} e a sua escrita na grid de armazenamento; em seguida (ii) K já se encontra em condições de ser usada para a encriptação de dados. O algoritmo que implementa este processo pode ser considerado em três fases: Inicialização, Dados I/O e Finalização, as quais são apresentados nas subsecções seguintes.

Inicialização

A primeira fase do algoritmo é dedicada à parametrização inicial do ambiente distribuído. As tarefas que descrevem esta fase de inicialização estão descritas sob a forma de diagrama de actividades na figura 3.

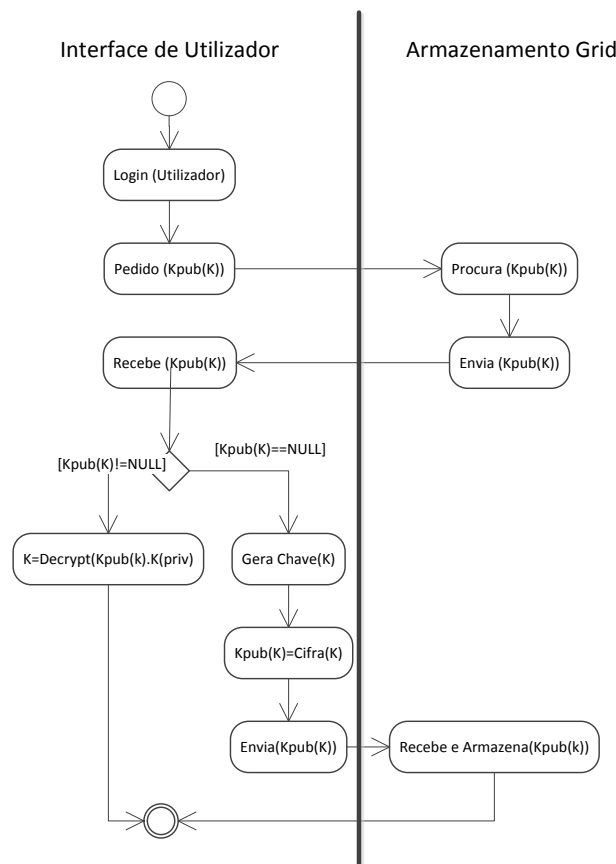


Figura 3 - Fase de Inicialização

Depois do utilizador entrar no ambiente da *grid* de armazenamento através da interface de utilizador, o algoritmo pede à *grid* de armazenamento a chave simétrica *Datakey* K encriptada pela chave pública do utilizador $K_{Pub}(K)$. Se a *grid* de armazenamento já foi inicializada, a sua resposta contém a chave encriptada *Datakey* $K_{Pub}(K)$ que será desencriptada pela chave privada K e gravada pelo interface de utilizador, numa zona de memória considerada segura. Caso contrário, no primeiro acesso ao armazenamento, a chave K tem de ser criada pelo algoritmo do interface de utilizador, sendo então encriptada e enviada para o armazenamento.

Dados I/O

A arquitectura organiza os dados guardados no armazenamento através de um sistema de ficheiros estruturado em directorias. Os dados são geridos e acedidos por um conjunto de primitivas como abrir, fechar, ler, escrever, eliminar, listar e renomear. Na figura 4, os algoritmos que implementam leitura, escrita e operações genéricas (eliminação, renomeação, listagem e outras) estão representados por diagramas de actividade. Em particular o algoritmo de leitura da figura 4(a) implica a desencriptação de dados recebidos pelo armazenamento, enquanto o algoritmo de escrita – figura 4(b) requer a encriptação dos dados antes de serem enviados para o sistema de armazenamento. As operações genéricas apenas enviam comandos ou sinais, como se pode verificar na figura 4(c).

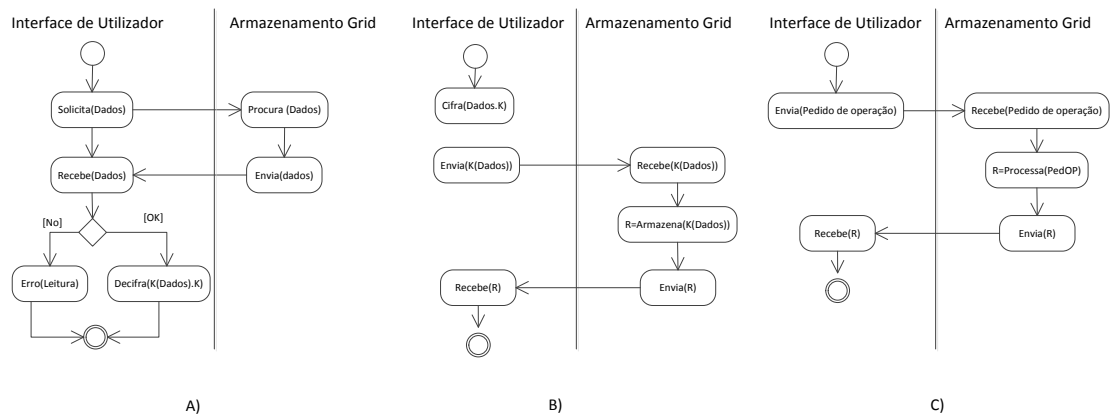


Figura 4 - Algoritmos de primitivas

Finalização

O algoritmo da fase de finalização é descrito pelo diagrama de actividades na figura 5. Antes do utilizador sair da *grid* de armazenamento, é necessário remover a chave

de dados simétrica e outra informação reservada da memória da interface de utilizador. Mas, como o utilizador pode ter ainda em curso operações activas de I/O sobre dados, é possível que pretenda saber o estado dessas operações e, desta forma, interroga a grid de armazenamento sobre as mesmas. Mediante a resposta, o utilizador pode escolher terminar a sua sessão ou esperar pela finalização das operações em curso. Finalmente, o utilizador sai da *grid* de armazenamento.

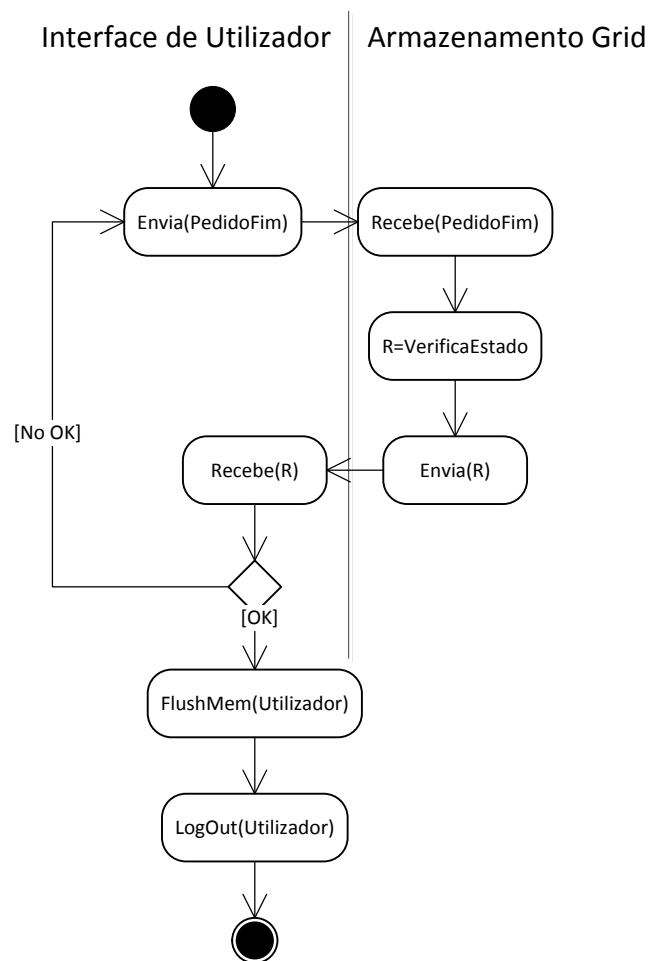


Figura 5 - Algoritmo e terminação

Capítulo 4 – Implementação do Sistema de Segurança de Armazenamento em Grid

A combinação de criptografia simétrica e assimétrica na segurança de dados descrita anteriormente, foi implementada como um serviço no *middleware* da *grid* de armazenamento *SRM*. Para descrever esta implementação apresentamos os constrangimentos e requisitos que motivaram as escolhas efectuadas e, em seguida, detalha-se a arquitectura do sistema de armazenamento e da sua biblioteca de funções.

Requisitos e Especificações

Como a implementação deste algoritmo tem de ser integrado num ambiente *SRM* que usa as suas próprias bibliotecas de armazenamento (GFAL), a melhor forma de simplificar o uso do armazenamento seguro da *grid* de armazenamento e obter uma melhor implementação no *middleware SRM* é usar a GFAL como base. Para obter um nível de alta segurança é também necessário que o serviço de segurança de armazenamento esteja disponível em modo interactivo na UI que efectua exclusivamente a descriptação de dados.

Nestas implementações, escolheu-se o algoritmo AES (Information, 2001) para a encriptação simétrica, e o algoritmo *Public Key Infrastructure* (PKI) (Rivest *et al.*, 1978 1978) para a criptografia assimétrica. Adicionalmente, para não sacrificar a simplicidade e portabilidade face a outros paradigmas, implementou-se uma interface Posix.

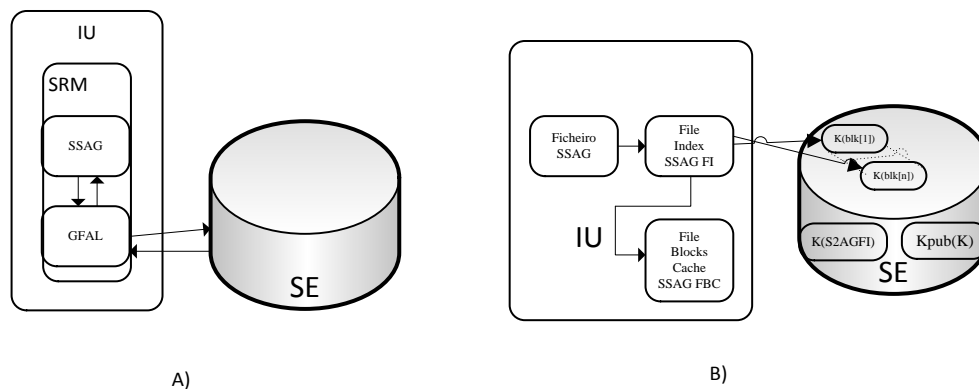


Figura 6 - a) Arquitectura e b) Sistema de Ficheiros

Arquitectura de Armazenamento

A arquitectura que implementa o algoritmo investigado no middleware SRM, que por sua vez satisfaz os requisitos descritos acima, está representada na figura 6(a). Assim o sistema de segurança é implementado como uma camada que trabalha em cima do GFAL, disponibilizando um serviço de ficheiros com capacidade de segurança e criptografia através da interface Posix.

O serviço de armazenamento desta arquitectura de segurança cria um sistema de ficheiros virtual estruturando os dados em ficheiros, directorias e subdirectorias, sem quaisquer restrições no número de ficheiros que as directorias podem conter. Como se construiu esta arquitectura sobre o GFAL, todos os objectos de dados são vistos como ficheiros guardados no SE, acessíveis aos utilizadores através da interface do GFAL (LFN, SRM, GUID e outros). Graças à arquitectura de armazenamento assim como à organização interna, esta implementação disponibiliza todos os benefícios do sistema de ficheiros. Um dos mais interessantes é a capacidade de modificação ou reescrita de ficheiros – operação não implementada pela biblioteca GFAL. Esta biblioteca apenas permite as operações de criação e escrita de novos ficheiros, sem quaisquer possibilidades de os modificar após a sua criação.

Um ficheiro pode ser guardado num SE num único bloco (*chunk*) com comprimento variável ou dividido por mais de um bloco de comprimento fixo – sendo este tamanho de bloco definido pelo utilizador na parametrização desta arquitectura de segurança, como mostrado na figura 6(b). Para evitar conflitos entre nomes de ficheiros, univocamente identifica-se cada bloco de dados (*chunk*) armazenado no SE através de um identificador UUID. O ficheiro índice mostrado na figura 6(b) (_FI) relaciona o ficheiro aos blocos correspondentes no SE. Este ficheiro índice é encriptado segundo o algoritmo simétrico e mantido em UI na memória estática. Desta forma, o utilizador usa um sistema de ficheiros virtual que não corresponde à sua estrutura física no SE, pois cada ficheiro pode ser dividido em diversos blocos guardados no SE como ficheiros. No entanto, o objectivo da indexação de ficheiros é a optimização das operações de I/O com a consequente redução do tempo de acesso aos dados. Como a reescrita e a modificação de ficheiros no SE tem de ser implementada através das primitivas do GFAL, estas operações são efectuadas apagando o ficheiro e reescrevendo a sua versão modificada. Assim, dividir um ficheiro em diversos blocos (*chunk*) no SE é a única forma de efectuar esta operação. O sistema de ficheiros é criado e armazenado no SE quando é efectuada a

inicialização desta arquitectura de segurança. Cada ficheiro com referência aos dados armazenados no SE é encriptado por uma chave simétrica guardada no mesmo SE e encriptada pela chave pública do utilizador.

No sentido de otimizar o desempenho das operações de I/O de um ficheiro, foi mantido na memória estática da UI uma *cache* local de partes de blocos (*_FBCC*). Todas as operações envolvendo partes de blocos já carregados na *cache* são executados localmente, variando o conteúdo de cada bloco. Quando um ficheiro é fechado, os blocos armazenados na *cache* são actualizados para a SE. Uma primitiva específica (*_flush*) é especificada para forçar o descarregamento dos dados da *cache* da UI para o armazenamento no SE. Isto aumenta notavelmente a velocidade de desempenho de um sistema de armazenamento, reduzindo o número de acessos ao SE. Problemas de coerência na *cache* podem surgir se houver mais de um acesso em simultâneo na *grid* de armazenamento a trabalhar sobre os mesmos dados. Foi aplicado um protocolo simples de consistência, permitindo ter diferentes cópias dos mesmos dados nas *caches* locais.

Biblioteca de Interfaces e API

Desde que a biblioteca de comandos implemente o Interface Posix.1, o acesso a um ficheiro num Sistema Virtual de Ficheiros encriptados é equivalente ao acesso a um ficheiro local. A arquitectura investigada especifica o mesmo conjunto de funções da biblioteca GFAL: no primeiro caso, as funções têm como prefixo *__** enquanto no segundo caso, começam por *gfal_**. A principal diferença entre a arquitectura e a interface Posix é a inicialização e a fase final, como está descrito na secção “Algoritmo”.

No parágrafo seguinte, especificamos as primitivas de inicialização da arquitectura apresentada através da caracterização das fases descritas anteriormente.

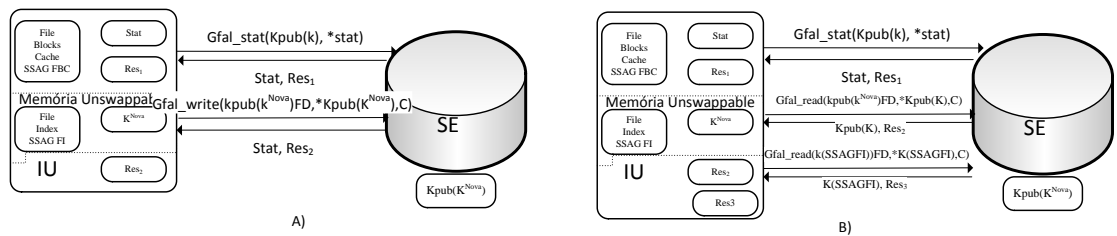


Figura 7 - Biblioteca de inicialização (a) primeira, (b) utilizações seguintes

Inicialização

A fase de inicialização é a mais importante fase da implementação. Nesta fase a biblioteca de contexto é inicializada através da activação das variáveis correspondentes às preferências do utilizador, tais como: `__PATH` (URL onde os ficheiros de dados vão ser guardados), `__PUBKEY` (chave pública que os utilizadores usam para encriptar), `__PRVKEY` (chave privada que os utilizadores usam para encriptar).

Um utilizador que precise de aceder ao SE necessita de invocar a função `__INIT`, para ler do espaço da *storage* uma chave K equivalente, encriptada pela chave pública do utilizador KPUB, como se mostra na Fig.7 e como se falou na subsecção “Inicialização”; os dois casos distinguem-se do primeiro pelos sucessivos acessos. Na primeira fase de inicialização, `__INIT` produz uma chave K simétrica como sequência de um intervalo de valores, devolvido pela função OPENSSL. Nos acessos seguintes a `__INIT` carrega a chave K e o ficheiro de indexação `__FI` dos elementos do *storage*.

Os algoritmos em ambos os casos são similares: inicialmente a UI verifica a presença de uma chave encriptada $K_{pub}(K)$ na SE através da função GFAL_STAT, depois, no caso de esta não existir, uma nova chave é criada (Fig.7a) e enviada para ao SE, caso contrário a chave e o ficheiro de indexação são carregados na UI através de duas operações consecutivas de leitura, através da função GFAL_READ (Fig.7b). A chave de dados encriptada $K_{pub}(K)$ é desencriptada através da chave privada do utilizador e colocada na memória local da UI para prevenir acessos indevidos.

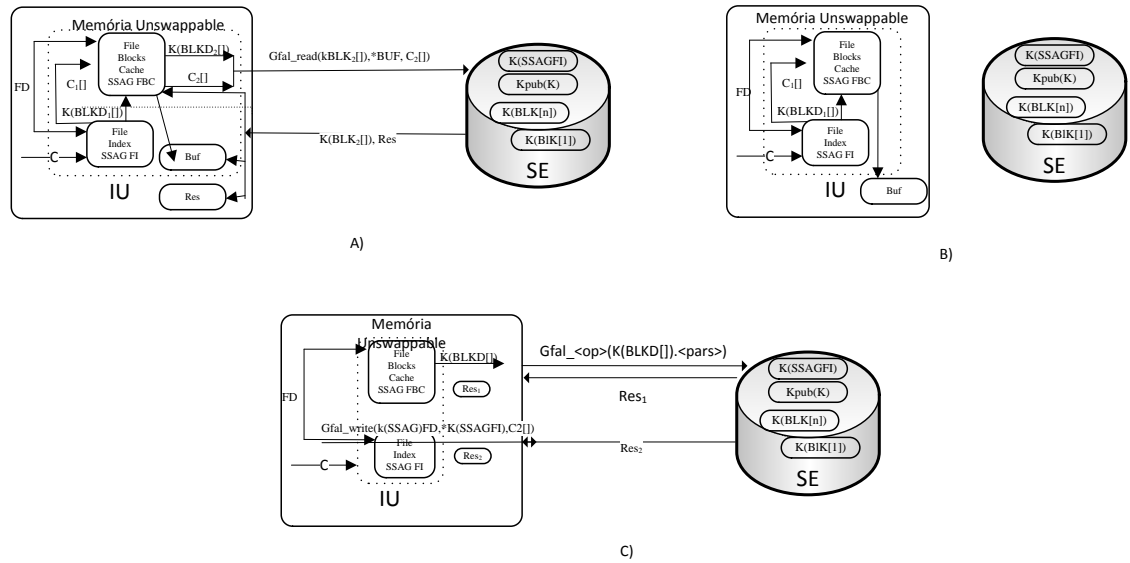


Figura 8 - Primitivas de Dados I/O : (a) __leitura, (b) __escrita, (c) __<op>

Dados I/O

As operações de I/O de dados são implementadas através de primitivas da I/O Posix tais como: abrir, ler/escrever e fechar. Os ficheiros são sempre encriptados na memória, a encriptação é efectuada em tempo real de execução. Para melhorar o desempenho da e a utilização da sua biblioteca, os ficheiros acedidos são localmente colocados na *cache* da UI até que sejam fechados. Quando o ficheiro é fechado, a *cache* da UI é sincronizada com a SE. Mais especificamente, a primitiva da função `__READ(int fd, void *buf, int c)` lê “c” bytes de dados do ficheiro referenciado no campo descrição do ficheiro fd, colocando-os no *buffer* buf da área local da UI. Tal como está ilustrado na Fig.7a, utilizando o ficheiro fd de indexação e os parâmetros de entrada, obtêm-se o correspondente conjunto de blocos descritores da SE (BLKD).

Os blocos que não estão na *cache*, identificados pelos conjuntos $BLKD2 \subseteq BLKD1$, são carregados da SE através da chamada da função `GFAL_Read`. Esses dados e os dados carregados da *cache* são colocados no *buffer* de saída, e a *cache* dos blocos e ficheiros é actualizada com os dados que acabaram de ser carregados da SE. Os conjuntos $BLKD1$ e $BLKD2$ correspondem aos vectores `BLKD1[]` e `BLKD2[]` da figura 8(a).

A função `__Write (int fd, const void *buf, int c)` é uma operação inteiramente executada localmente na UI, como mostra a figura 8(b). Os blocos de dados que vão ser

modificados na SE são temporariamente guardados na *cache* de blocos de ficheiros. Quando o ficheiro é fechado, renomeado, movido, eliminado, a limpeza da *cache* é forçada ou a sessão *SRM* é terminada. Os dados da *cache* são sincronizados com os seus correspondentes no SE.

__<op> (int fd, <par>) é uma operação genérica de I/O que corresponde à operação GFAL – gfal_<op> (int fd, <par>). Quando a __<op> (int fd, <par>) modifica a estrutura do sistema de ficheiros (eliminar, renomear, mover, *mkdir* e outros) é necessário actualizar o ficheiro de indexação do SE.

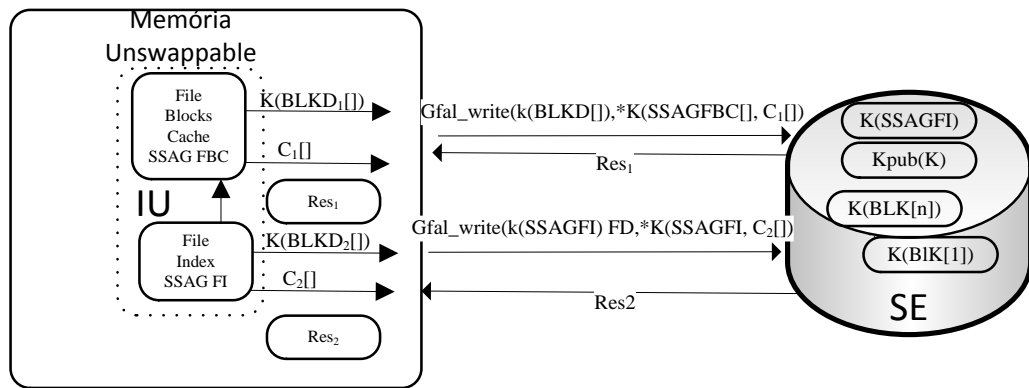


Figura 9 - Implementação da Primitiva de Finalização

Finalização

O objectivo principal da operação de Finalização é a sincronização dos dados entre a *cache* da UI e a SE. É implementado pela função __Finalize(), uma versão simplificada daquilo que está detalhado na Figura 9. Aqui descrevem-se 2 operações diferentes de gfal_write dentro do SE. A primeira escreve todos os dados que estão nos blocos de ficheiros da *cache* da UI (_FBC), a outra escreve o ficheiro de indexação da UI (_FI).

Esta sequência implementa a função __Flush, chamada sempre que um ficheiro é fechado, eliminado ou renomeado. Além disso, isto é uma versão simplificada da __flush, visto que as bibliotecas GFAL não implementam a capacidade de reescrita: um ficheiro só pode ser escrito quando é criado. Assim, se um ficheiro já existe no SE, a GFAL não permite modificá-lo. No sentido de implementar esta capacidade utilizando a biblioteca GFAL, é necessário ultrapassar o problema da reescrita apagando e criando o novo ficheiro cada vez que um ficheiro é modificado. Contudo, o algoritmo de reescrita foi inteiramente implementado na biblioteca Teste.

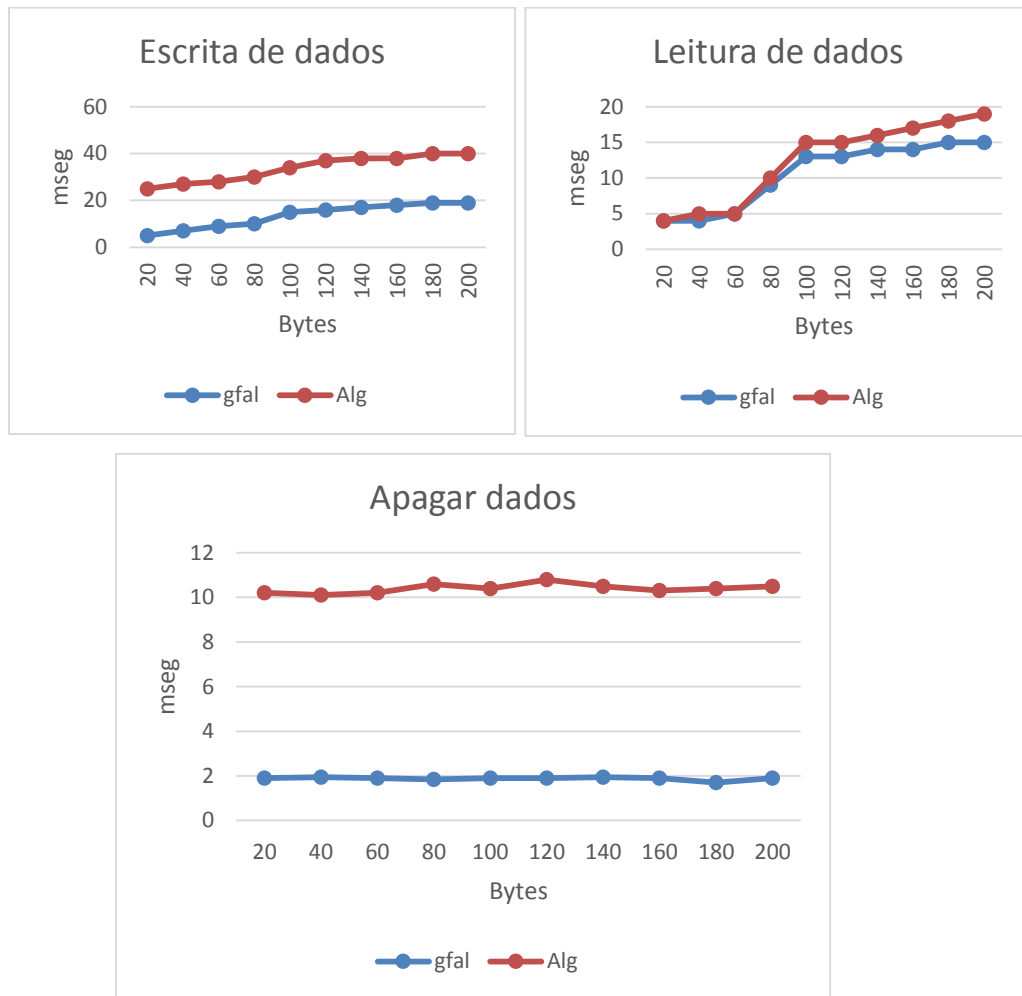


Figura 10 - Desempenho das operações

Desempenho

Foram executados testes para avaliar o desempenho da arquitectura investigada (Figura 10). Nestes testes, foi criado/escrito um ficheiro e sobre o qual foram feitas operações de leitura e eliminações. Como na execução se foi variando o tamanho do ficheiro de 28 até 217 bytes e duplicando o tamanho em cada experiência, pelo que no total foram executados 10 testes diferentes. Através deles, avaliámos o desempenho das primitivas da arquitectura implementada. O comportamento tem sido comparado com o comportamento da GFAL. No sentido de providenciar um enquadramento completo do desempenho da arquitectura invstigada, tomámos as mesmas medidas no sistema de ficheiros local.

Nos testes, avaliámos o desempenho das operações executadas directamente na SE considerando diferentes ambientes. Isto pode ser tido como o pior caso para a arquitectura investigada, em que cada operação é directamente sincronizada com o SE,

sem chegar a ir à *cache*. Como métrica de desempenho, considerámos o tempo de resposta como o tempo que passou entre as operações de lançamento até aos resultados para o utilizador. No sentido de obter dados significativos, repetimos cada teste 1000 vezes e calculámos o intervalo dos resultados obtidos.

Escrita:

O resultado obtido por avaliar o tempo de resposta das chamadas de criação/escrita são visíveis na figura 10(a). Estes resultados mostram padrões similares para todo o ambiente considerado. Tal como era esperado, os tempos de resposta de operações de escrita do UI para o SE são afectados pelo tamanho do ficheiro. A transferência e armazenamento dos dados dificilmente são afectados. Comparando os resultados de diferentes bibliotecas, podemos observar que, não considerando o impacto da *cache*, a arquitectura investigada é consideravelmente mais lenta que a GFAL e obviamente que as chamadas locais.

Isto deve-se ao facto de, em cada vez que se escreve no SE, é também necessário actualizar o ficheiro de índice e, consequentemente, duas operações `gfal_write` são necessárias, como se mostra a figura 9. Mas, como podemos reparar observando o desempenho da GFAL, o tempo gasto no acesso à rede de comunicação é exponencialmente maior do que o tempo de processamento gasto na encriptação dos dados. Isto justifica a diferença de desempenho entre esta arquitectura e as outras. No primeiro caso, são necessários 2 acessos à rede no armazenamento dos dados: o primeiro para guardá-los e o segundo para actualizar o ficheiro de índice, enquanto nos outros casos apenas um acesso é necessário. Isto é o custo da reescrita: para implementar uma funcionalidade tão importante, a arquitectura investigada introduz a fragmentação de ficheiro e, consequentemente, uma tabela de indexação.

Leitura:

O desempenho obtido por testes de leitura podem ser visualizados na figura 10(b). Similarmente à operação de escrita, os tempos de resposta aumentam quando aumenta a quantidade de dados devido à necessidade de passar esses dados através da rede de comunicação. Mas, neste caso, os resultados do pior caso `__read` são comparáveis ao `gfal_read`. Isto deve-se ao facto de numa operação de leitura não ser necessário fazer qualquer *update* ao índice e a sobrecarga de processamento deve-se essencialmente à

encriptação. Os padrões também mostram que o tempo gasto nas tarefas de encriptação é negligenciável relativamente ao tempo gasto na comunicação dos dados.

Eliminação:

A figura 10(c) mostra os resultados obtidos pelas operações de eliminação. Obviamente, o desempenho das operações de eliminação não varia com o tamanho do ficheiro, uma vez que apenas alguns *bytes* são enviados. Tal como no caso da escrita, também aqui existe uma grande diferença de desempenho entre a arquitectura investigada e as outras devido aos mesmos motivos: como mostra a figura 8(c), a eliminação de um ficheiro necessita de actualizar o ficheiro de índice depois de eliminar o ficheiro do SE. Isto introduz mais uma operação *gfal_write*, aumentando o tempo total de uma operação de remoção. Pode ser também notado que o desempenho da operação de remoção não varia com o tamanho do ficheiro.

Capítulo 5 – Avaliação

Quer a abordagem proposta quer a implementação foram baseadas na ideia de fornecer um sistema de ficheiros seguro em vez de ficheiros seguros, combinando cifragem simétrica e assimétrica. Esta abordagem tem várias vantagens:

1. Segurança – cifrando a *datakey* simétrica pela chave pública do utilizador assegura que os dados e a *datakey* são exclusivamente acessíveis apenas pelo utilizador/dono autorizado.
2. API de bibliotecas de Interfaces – Um conjunto completo de bibliotecas que introduz novas capacidades (por exemplo, modificação de ficheiros, reescrita e renomeação) e optimizando a já existente por visar a segurança.
3. Desempenho – A arquitectura foi desenhada para satisfazer necessidades específicas de desempenho como indexação de ficheiros e *cache* local.
4. Fiabilidade e tolerância a falhas – uma vez que a *datakey* não está distribuída por diferentes nós, é possível implementar sistemas de armazenamento fiáveis e tolerantes a falhas.

Esta abordagem tem também algumas desvantagens:

1. Segurança – ninguém pode aceder aos conteúdos dos ficheiros, mas os administradores podem apagá-los.
2. Sobrecarga – manter a estrutura do sistema de ficheiros tem a desvantagem de que, cada vez que um ficheiro é fechado, renomeado ou apagado, a estrutura do sistema de ficheiros tem de ser actualizada, introduzindo tempo extra nas operações de escrita necessárias.
3. Problema de consistência – a presença de *cache* pode introduzir problemas de consistência de dados no caso de diferentes acessos e modificações simultâneas nos mesmos ficheiros.

De qualquer forma, estas desvantagens não afectam ou comprometem a validade da abordagem que é um contributo significativo para as actualizações mais recentes relatadas na secção introdutória e que sobressai pelas suas vantagens. Além disso, constitui uma base para trabalho futuro.

Capítulo 6 – Conclusão

Neste trabalho descrevemos um sistema de segurança de armazenamento em *Grid*, um sistema de armazenamento seguro (cifrado) para armazenamento distribuído (*grid* de armazenamento). Foi implementado e integrado com o *middleware SRM*. Detalhámos o algoritmo de segurança e a implementação da arquitectura. O algoritmo de segurança é baseado no conceito da combinação de criptografia simétrica e assimétrica. A criptografia simétrica é directamente aplicada aos dados, proporcionando armazenamento de dados cifrados. A chave simétrica que decifra os dados é por seu turno cifrada pela chave pública do utilizador proprietário desses dados (criptografia assimétrica) e guardada no sistema de armazenamento distribuído. A decifragem é efectuada pelo nó que faz o interface com o utilizador e ambas as chaves e os dados são colocados em memória volátil não partilhada ou paginada nesse nó. Desta forma os dados podem ser acedidos exclusivamente pelo seu proprietário. Para poder partilhar esses dados com outros utilizadores é necessário guardar no sistema distribuído cópias das *DataKeys* cifradas pela chave privada de cada utilizador.

O ponto forte da implementação da arquitectura investigada no *middleware SRM* é a definição de um sistema de ficheiros seguro a correr sobre a biblioteca GFAL. Esta escolha permite proteger tanto os dados/ficheiros como também a sua estrutura. Além disso, a implementação sobre o *SRM* introduz uma nova capacidade: alteração e reescrita de ficheiros. Esta implementação foi avaliada particularmente em relação a 3 operações principais: leitura, escrita e eliminação. Os testes foram feitos considerando o pior caso, em que na arquitectura investigada se opera sempre directamente com o SE, permitindo a comparação com as outras bibliotecas (GFAL e LOCAL). Os resultados obtidos mostram tempos de resposta mais altos na arquitectura investigada em relação às outras, nas operações de escrita e eliminação devido à operação de escrita no ficheiro de índice que cada uma destas operações implica, enquanto a operação de leitura está a par com a GFAL. Uma investigação mais aprofundada do desempenho da arquitectura investigada considerando o impacto da *cache* será um dos possíveis desenvolvimentos futuros. Outros pontos interessantes para se investigar futuramente são os seguintes: melhorias de segurança, coerência de *cache*, partilha de dados, tolerância a falhas, qualidade do serviço (QoS), optimização do sistema e processos *batch*.

Parte II

Capítulo 7 – Curriculum

7.1. Dados Pessoais

Nome: Francisco José Fialho Nunes

Idade: 47 anos

Data de nascimento: 1 de Abril de 1967

Nacionalidade: Portuguesa

Estado Civil: Solteiro

Bilhete de Identidade: N° 7750663, emitido em 5/04/2004 pelo arquivo de identificação de Lisboa.

Contribuinte: 184630754

Morada: Rua das Torres, 7, 2-Esq

Alfragide

2610 - 175 Amadora

Telf.: 96 6918 883

7.2. Habilitações Literárias

7.2.1. Formação Universitária

- Licenciatura em Matemática (1986-1991), pela Universidade de Évora.
- MBA, ISET – Porto (1994)
- Licenciatura em Engenharia Informática (1999-2004), Universidade Autónoma de Lisboa

7.2.2. Formação Complementar

Ano 1986

- Frequência de um curso de Introdução ao BASIC, na delegação de Évora do sindicato dos bancários do Sul e Ilhas.

Ano 1987

- Frequência de um curso de Complementos de Basic, na sequência do curso do ano transacto.
- Introdução às Redes Novell

Ano 1988

- Frequência num Seminário sobre novas Tecnologias, promovido pelo Ministério da Indústria e Energia, de onde se destacam dois de vários pontos abordados:
- Burótica
- Telecomunicações
- Frequência do Curso de Tecnologia Educacional, promovido pelo INSTITUTO DE TECNOLOGIA EDUCATIVA.
- Introdução à programação (Pascal)

Ano 1989

- Frequência do Curso de Novas Técnicas de Gestão de PME's Comerciais, promovido conjuntamente por Centro de Formação Profissional para o Comércio de Afins (CECOA), Instituto de Emprego e Formação Profissional (IEFP), Confederação do Comércio Português (CCP), o qual foi ministrado nas instalações da Associação Comercial do Distrito de Évora. Neste curso importa destacar a Aplicação de velhos conceitos a novas tecnologias, nomeadamente a aplicações informáticas, tais como:

- Processamento de texto
- Folha de cálculo
- Base de Dados
- Aplicação de processamento de Salários
- Aplicação de Gestão de Stocks
- Aplicação de Facturação
- Aplicação de Contabilidade
- Aplicação de Gestão Integrada

Ano 1990

- Frequência de um módulo de Filosofia da Programação, promovido pela Norma.
- Programação científica (FORtran)

Ano 1991

- Frequência do Curso de Formação Pedagógica de Formadores, ministrado na UNESUL.

Ano 1992

- Frequência de um Curso de Formação sobre Conceitos e Práticas de Avaliação, Tipos de Instrumentos de Avaliação, Estrutura Modular Avaliação e Progressão Individualizada, ministrada na EPBJC.
- Frequência de um Curso de Telemática promovido pelo pólo distrital de Évora do projecto Minerva.

Ano 1993

- Frequência e conclusão do Módulo I de IV do CESE “Gestão e Administração Escolar”.

Ano 1994

- Frequência e conclusão do Módulo II de IV do CESE “Gestão e Administração Escolar”.
- Frequência, com aprovação de Muito Apto, do curso de Socorrismo Essencial promovido pela Cruz Vermelha, Delegação de Évora.

Ano 1995

- Frequência e conclusão do Módulo III de IV do CESE “Gestão e Administração Escolar.

Ano 1996

- Frequência na área de IBM, dos seguintes cursos:
 - AS/400 Para Novos Utilizadores
 - AS/400 Operadores de Sistema
 - AS/400 Facilidades do Sistema
 - AS/400 Bases de Dados

Ano 1997

- Frequência na área de IBM, dos seguintes cursos:
 - AS/400 Internet Access e TCP/IP
 - AS/400 Client Access
 - Administração de Sistemas AIX/RS6000
 - OS/390 - Auditores
 - OS/390 - JCL utilities
- Frequência noutras áreas, dos seguintes cursos:
 - Auditoria Técnica Informática
 - Metodologia e Gestão de Projectos
 - Planeamento e Instalação DOMINO em AS/400 e Windows NT
 - Analistas/Programadores

- Administração de Bases de Dados

Ano 1998

- Frequência na área de IBM, dos seguintes cursos:
 - ISPF
 - DB2
 - CICS/Cobol/DB2
 - JES
 - Spooler

Ano 1999

- Frequência na área de IBM, dos seguintes cursos:
 - Novos meios de formação: e-Learning
 - MindSpan

Ano 2000

- Frequência na área de IBM, dos seguintes cursos:
 - Novos meios de formação: e-Learning
 - Metodologia IBM de conversão para introdução a moeda Euro

Ano 2001

- Computação Distribuída (Arquitectura de sistemas e Programação em Java)
- Formação JAVA em IBM Domino
- Formação em Administração em Informix
- Programação em 4GL

Ano 2002

- Kickoff IBM Systems and Technology Group – Nice
- Processos de Produção – Fábrica IBM em Montpellier
- Escola de vendas IBM - modelo SSL
- Segurança Informática – Modelos de Implementação
- Segurança Informática – Metodologias de Auditoria e Consultoria

Ano 2003

- Kickoff IBM Systems and Technology Group – Glasgow
- Consolidação de **Infra-estrutura** Tecnológica
- Particionamento lógico de **Infra-estrutura** – Futuro Tecnológico
- Novas Necessidades em Ambientes de Segurança
- Seminário INTERNET -> HyperNET (Don TapScott)

Ano 2004

- Kick-off IBM Systems and Technology Group – Orlando (Florida)
- Virtualização, níveis de abstracção de modelação de negócio
- **Infra-estruturas** tecnológicas como base estratégica
- Seminário IBM Academy of Technology (Madrid) – speakers et al 2 Nobel Prizes
- Formação específica em SAN Volume Controller
- Virtualização de dispositivos – novos paradigmas
- Algoritmos de virtualização de sistemas de armazenamento
- Protocolos de comunicação de dados
- Computação ubíqua e Internet das coisas

Ano 2005

- Kick-off IBM Systems and Technology Group – Madrid
- Sistemas de armazenamento de dados classe empresarial
- Metodologias de Business Continuity and disaster Recovery

- Gestão de Projectos
- *Grids* de computação e de Armazenamento
- Protocolos internos de comunicação de dados em sistemas de armazenamento
- Architecturas IT

Ano 2006

- Kick-off IBM Systems and Technology Group – Lisboa
- Liderança
- Motivação
- Gestão de Equipas

Ano 2007

- Kick-off IBM Systems and Technology Group – Paris
- Grupos de serviços partilhados – Integração e gestão de equipas

Ano 2008

- Kickoff Compta - Lisboa
- Outsourcing: Alocação e optimização de recursos

Ano 2009

- Kickoff Compta - Lisboa
- Seminário APDC
- Métodos de Investigação em Ciências e Tecnologias da Informação I (9 Créditos) - ISCTE
- Métodos de Investigação em Ciências e Tecnologias da Informação II (3 Créditos) - ISCTE

Ano 2010

- Kickoff Compta - Lisboa
- Seminário ANETIE
- Projecto de Investigação em Ciências e Tecnologias da Informação I (9 Créditos) - ISCTE
- Projecto de Investigação em Ciências e Tecnologias da Informação II (2 Créditos) – ISCTE
- Sistemas em grelha (UC do mestrado em EI - UAL)
- Integração de Sistemas (UC do mestrado em EI - UAL)

Ano 2011

- Kickoff Compta - Lisboa
- Seminário em Metodologias de Investigação (UAL)
- Projecto de Investigação em Ciências e Tecnologias da Informação III (2 Créditos) - ISCTE
- Projecto de Investigação em Ciências e Tecnologias da Informação IV (2 Créditos) - ISCTE

Ano 2012

- Kickoff Compta - Lisboa
- Seminário APDC
- ITIL Foundations
- PMI
- Seminários de Investigação e Comunicação em Ciências e Tecnologias da Informação I (4 Créditos) - ISCTE
- Seminários de Investigação e Comunicação em Ciências e Tecnologias da Informação II - ISCTE

7.3. Experiência Pedagógica

7.3.1. Área Educacional

- Exerceu funções docentes no ensino secundário desde o ano lectivo de 1985/86 até ao ano lectivo 1996/1997.
- Desde 2006 na UAL a leccionar diferentes disciplinas (GSR, SI, ER, ES)

7.3.2. Área de Informática

Ano 1987-88

Teve a cargo o departamento de formação na SPRI-4, durante estes dois anos, responsabilizando-se pela estrutura, projecto e avaliações de acções de informática das quais se destacam:

Introdução aos Computadores

Arquitectura de Computadores

Sistema Operativo (MS-DOS)

Tratamento de texto (Wordstar P4.0)

Base de Dados

Folha de Cálculo

Filosofia da Programação

BASIC (int.)

BASIC (comp.)

Ano 1989

Projetou e elaborou uma acção de Introdução à Informática na delegação de Évora do Ministério da Indústria e Energia, secção de Metrologia.

Ano 1990

Ministrou, a convite da Unesul, o módulo de Informática de Gestão no âmbito do Programa FIQ's 90.

Ano 1991

Ministrou o mesmo módulo relativo ao ano transacto do programa FIQ 91. Ministrou o módulo de Introdução à Informática inserido no programa IJOVIP 91.

Ministrou o módulo de Introdução à Informática do Curso de Gestão de Aprovisionamentos.

Ano 1992

À imagem dos anos transactos continuou a formação do módulo de Informática de Gestão no programa FIQ's 92, acrescentando o módulo de Harvard Graphics.

Coordenou e Ministrou uma acção de Aplicações de Micro-Informática para funcionários da Unesul.

Ministrou um curso de APLICAÇÕES WINDOWS no Núcleo Empresarial da Região de Évora (NERE), em que se destacam:

Ambiente Windows (3.1)

Tratamento de texto (Windows 2.0)

Folha de Cálculo (Excel 3.0)

Base de Dados (Super Base 4 1.0)

Ministrou um Curso de Microinformática na União de Sindicatos do Distrito de Évora.

Ano 1993

Ministrou um Curso de Técnicas de Programação e Programação em DBASE III Plus.

Ano 1994

Ministrou várias acções de curta duração (30h) de Iniciação à Microinformática em Ambiente Windows.

Ano 1995

Maio

Ministrou, projetou e coordenou um Curso Técnico de “*Hardware Elementar*” (Organização *Softline*).

Junho

Ministrou, projetou e coordenou um Curso de Introdução à Informática para colegas da Escola Profissional Bento de Jesus Caraça.

Ministrou uma acção de formação de curta duração (30h), na Softline, no âmbito do ambiente Windows, Processamento de Texto (Microsoft Word 6.0) e Folha de Cálculo (Excel 5.0)

Monitor do Módulo de Informática (180h), do Curso de Técnicos Administrativos - CGTP-IN (Instalações da Softline).

Setembro

Monitor do Módulo de MS-DOS (24h) do Curso de Informática para Administrativos Promovido pelo Sindicato dos Trabalhadores da Função Pública do Sul e Açores, ministrado nas instalações da Softline.

Outubro

Monitor do Módulo de Windows 3.11 (24h) do Curso de Informática para Administrativos Promovido pelo Sindicato dos Trabalhadores da Função Pública do Sul e Açores, ministrado nas instalações da Softline.

Novembro

Monitor do Módulo de Microsoft Word 6.0 (32h) do Curso de Informática para Administrativos Promovido pelo Sindicato dos Trabalhadores da Função Pública do Sul e Açores, ministrado nas instalações da Softline.

Ano 1996

Instrutor e Coordenador de Formação na área de Mid Range Servers na IBM.

Ano 1997

Continuou a sua acção de Coordenação e Instrução de Formação na IBM.

Coordenação de desenvolvimento de um auto-estudo e-learning para a Tranquilidade Seguros.

Planeamento de novos desenvolvimentos de formação em auto-estudo.

Ano 1998

Continuou a sua acção de Coordenação e Instrução de Formação na IBM.

Implementação de processos de e-Learning massificados em grupos bancários.

Ano 1999

Continuou a sua acção de Coordenação e Instrução de Formação na IBM.

Modelos de formação em open systems

Modelos de migração de dados – ANO 2000

Ano 2000

Continuou a sua acção de Coordenação e Instrução de Formação na IBM.

Formação de quadros do grupo Cofinoga Internacional – Maginfo (França), SDDC (França), SIMEON (Espanha), Credibra (Brasil)

Modelos de Migração de Dados Conversão Escudos - Euro

Ano 2001

Continuou a sua acção de formação interna do Grupo Cofinoga (Domino/Notes, MQSeries, AIX...)

Ano 2002

Formação de Parceiros IBM em Systems Storage

Evento anual de parceiros 2002 (Plateia de 150 pessoas)

Divulgação do brand de storage para clientes

Ano 2003

Continuação de Formação de Parceiros IBM em Storage,
Consolidação e Virtualização (acções múltiplas ao longo do ano).

Evento anual de parceiros 2003 (Plateia de 300 pessoas)

Divulgação do brand de storage para clientes (acções múltiplas ao
longo do ano)

Speaker no seminário Grandes Sistemas – Evento Jornadas
Informáticas Universidade do Minho

Speaker no seminário de Segurança – Evento TechData

Ano 2004

Continuação de Formação de Parceiros IBM em Storage,
Consolidação e Virtualização (acções múltiplas ao longo do ano).

Evento anual de parceiros 2003 (Plateia de 450 pessoas)

Formação Interna para IBMers (Segurança, Virtualização,
Consolidação, Infra-estrutura tecnológica)

Divulgação do brand de storage para clientes (acções múltiplas ao
longo do ano)

Speaker no seminário de Techconnect, afiliado IBM Academy of
Technology (GTO 2004)

Speaker no seminário de Inovação Tecnológica – Evento TechData

7.4. Experiência Profissional

Ano 1985

Iniciou a sua carreira profissional no ano lectivo 85/86 onde
exerceu funções docentes na Escola C+S de Arraiolos até ao ano lectivo de
88/89. De 89/90 até 92/93 na Escola C+S André de Resende e do ano

lectivo 93/94 na Escola Secundária André de Gouveia de Évora. Terminou esta actividade em Dezembro de 1996.

Ano 1987

No biénio 87/88 colaborou na SPRI-4, uma empresa de serviços onde prestava todo o apoio de direcção tecnológica na área de informática.

Ano 1993

No ano de 1993 foi convidado a integrar uma equipa de trabalho no departamento de informática em Administração de Sistemas daquela entidade.

Ano 1994

Em 1994 foi convidado, pela CARDITA, a integrar uma equipa nacional de consultores num *software* inovador para elaboração de horários escolares com o GP-Untis (*Software* Aplicacional).

Ano 1995

Em 1995 foi convidado a integrar a Direcção da Delegação de Évora da Escola Profissional Bento de Jesus Caraça.

Em 1995 foi responsável pelo estudo e implementação do *software* e *hardware* de suporte à rede informática, central, do Ministério das Finanças de Cabo Verde (Novell 3.12).

Foi também responsável pela configuração e operacionalização do sistema de suporte da rede de Informática do Ministério das Finanças de Cabo Verde.

No mesmo ano, no âmbito deste trabalho de consultoria, definiu os procedimentos diários a ser efectuados pelos operadores, nomeadamente cópias de segurança e operações de consolidação de dados entre as várias Repartições de Finanças e os Serviços Centrais da Direcção Geral das Contribuições e Impostos (DGCI).

Autor de um guia prático de operação com redes locais (LAN-Novell).

Ano 1996

De Janeiro a Dezembro de 1996 exerce funções de Director de Informática na organização “Bolas, Máquinas e Ferramentas de Qualidade”, sediada em Évora, com delegações em Lisboa e Porto, onde tinha a reportar 8 pessoas. Os principais projecto que levou a cabo foram:

Manual de controlo de gestão automatizado e de disponibilização JIT (Trabalho em VB)

Integração da rede móvel da equipa de vendas com mobilidade nacional com computação móvel em inter-acção com o sistema central AS/400

Ano 1997

De Janeiro de 1997 até Junho de 2000 integrou os quadros da Companhia IBM Portuguesa, como IT Coordinator na área de Servers, com 4 pessoas a reportar, onde esteve envolvido em projectos tais como:

Projecto de migração de ambiente Apple para ambiente Windows NT na KPMG

Projecto de implementação de AS/400 no Banco Finantia

Projecto de implementação de AS/400 na TMG

Projecto de migração de MVS para OS/390 da Portugal Previdente

Projecto de implementação RS/6000 SPII na SONAE

Projecto de implementação RS/6000 SPII na Mello/Império

Projecto de implementação SAP em AS/400

Projecto de implementação de Domino em AS/400

Projecto de implementação RS/6000 SPII na Servibanca

Projecto de desenvolvimento CICS/COBOL/DB2 na P. Previdente

Projecto de desenvolvimento CICS/COBOL/DB2 no BCP

Projecto de implementação e desenvolvimento de MQSeries, WebSphere na SONAE

Projecto de implementação e desenvolvimento de MQSeries, WebSphere na Mello/Império

Projecto de migração e desenvolvimento de VM/VSE para OS/390 na DAMAG (Marinha de Guerra Portuguesa)

Projectos de implementação e-business

Projectos de implementação de e-learning

Projecto de implementação de Lotus Learning Space, como base de projectos de e-learning

Projecto de parceria com a NetGen, uma empresa Norte Americana, para implementação de projectos de e-learning.

Ano 2000

De Junho de 2000 a Março de 2001 integrou o Grupo Cofinoga, como Sub-Director na Área de Exploração, ficando com 20 pessoas a reportar directamente, onde planeou, coordenou e geriu o projecto de trabalho numa estratégia de desenvolvimento e consolidação através de migração de servidores (INTEL, RS6000, AS/400, SUN Fire e S/390) e comunicações de voz e dados entre os três pontos principais Lisboa, Porto e Paris, bem como a infra-estrutura da rede de parceiros de cartões de crédito de redes privadas, um pouco por todo o país, donde se destaca o Grupo Auchan.

Ano 2001

Após o projecto Cofinoga regressou à IBM em Abril de 2001 onde foi responsável pelo desenho e arquitectura de soluções na área de storage, interface com clientes e gestores de projecto. Desempenha a função de consultor em sistemas de segurança e integra projectos onde elabora levantamento de requisitos, planeia e desenha soluções de storage, backup e de disaster recovery. Integra ainda projectos onde elabora levantamento de requisitos, planeamento e desenho de soluções funcionais e de negócio. Excedeu objectivos na avaliação de desempenho no último ano.

Ano 2007

Em 2007, ao abrigo do Programa IBM GROWTH, mantendo o vínculo laboral com a IBM vai para a Compta criar uma nova parceria IBM e desenvolver o negócio IBM na COMPTA. Esta parceria revestiu-se de um enorme sucesso pois a Compta é hoje o maior parceiro IBM em Portugal, em Cabo Verde e um dos maiores em Angola. Na Compta liderou a reestruturação da empresa tornando-a uma empresa mais adequada ao mercado nacional. Em 2008 e 2009 desenvolveu iniciativas junto das entidades académicas e formou parcerias de desenvolvimento que resultaram na criação de uma unidade de I&D na Compta. Esta unidade tem tido alguns êxitos sendo financiada em grande parte dos projectos, por fundos de apoio tais como o QREN. Em 2010 tem as primeiras experiências internacionais e percorre o mercado Africano, Sul Americano e Asiático numa tentativa de criação de uma estratégia para o desenvolvimento internacional de negócio do Grupo Compta, nunca perdendo de vista o desenvolvimento de negócio IBM. Em 2011 centra-se nos mercados cabo verdeano, angolano e moçambicano. Em 2012 o mercado cabo verdeano está consolidado e dedica 2013 à consolidação do mercado angolano. Está previsto o regresso à casa mãe IBM em Setembro de 2014.

7.5. Publicações**7.5.1. Capítulos de livros**

- AS/400 Internet Security: IBM Firewall for AS/400, IBM Redbooks, 16 de Julho 1998, ISBN 0738400254 , IBM Form Number SG24-2162-00
- “Fórum Armazenamento, Semana Informática” 591, pag. 11, March 8 2002
- “Empresas Optimizam Informação” – ILM, Semana Informática 748, pag.10, 24 de Junho de 2005

7.5.2. Publicações de trabalhos completos em conferências

- Nunes, F.; Falcão, S.; Carneiro, A. - "Managing Storage Systems in an Educational Environment"; Proceedings of V International Conference On Multimedia and Information and Communication Technologies in Education, 22-24 April 2009, Lisbon, Abstract code 560, book 1, pag 89-94, Proceedings of m-ICTE 2009, 2009.
- Nunes, F.; Falcão, S.; Carneiro, A. - "Application of Lean Six Sigma Methodology to Optimization Process of Data Management in an Educational Context"; Poster in KMIS 2009, Madeira, 6 - 8 October, 2009, Portugal
- Nunes, F.; Falcão, S.; Carneiro, A. - " Lean Six Sigma na Gestão de Informação em Entidades Educacional"; Poster in 2nd Portuguese Six Sigma Conference – “The Power of Knowledge”; 23-24 Outubro, Tomar, 2009.
- Nunes, F.; O’Neill, Henrique – “Rumo a um framework de gestão de sistemas de armazenamento na banca em Portugal”; Doctoral Consortium CAPSI 2009, 28-30 Outubro 2009, Viseu
- Nunes, F.; O’Neill, Henrique – “A próxima geração de arquitecturas de armazenamento de dados no sector financeiro”, Proceedings of X CAPSI, 20-23 Outubro 2010, Viana do Castelo, 2010.
- Nunes, F.; O’Neill, Henrique – “Storage Resource Management em sistemas de armazenamento como suporte a *grid*s de serviços para o sector financeiro”, Proceedings of Conferência IADIS Ibero Americana WWW/INTERNET 2010, 10 – 11 de Dezembro 2010, Algarve, Portugal, Abstract code S125_pt, book 1, pag 232-233, Proceedings of IADIS 2010, 2010
- Nunes, F.; Falcão, S. - “Lean Six Sigma applied to quality and productivity improvement in the managing cycle – planning and control – of an education, training and social integration institution”; Proceedings of KMIS 2011, 26-29 October 2011, Paris, France (Paper accepted)
- Nunes, F; O’Neill, H. – “Mecanismos de gestão de autorizações em ambiente de *grid* de armazenamento”, In Proceedings of 11ª Conferência da Associação Portuguesa de Sistemas de Informação, 19 a 21 de Outubro de 2011, ISEG - Instituto Superior de Economia e Gestão, em Lisboa, numa

organização conjunta com o IST - Instituto Superior Técnico e ISCTE - Instituto Universitário de Lisboa., Portugal

- FALCÃO, S.; NUNES, F. *et al.* - Lean Six Sigma applied to quality and productivity improvement in the management cycle – planning and control – of an education, training and social integration institution. In International Conference on Knowledge Management and Information Sharing –KMIS 2011, Paris, 2011. ISBN 978-989-8425-81-2. p.326-330
- Nunes, F; O’Neill, H. – “Sistemas de autorização baseados em confiança em ambientes federados aplicado aos serviços de saúde”, In Proceedings of 12^a Conferência da Associação Portuguesa de Sistemas de Informação, Outubro de 2012, Universidade do Minho, em Guimarães, UM
- Nunes, F; O’Neill, H. – “Framework de controlo de acesso baseado em variação”, In proceedings of 13^a Conferência da Associação Portuguesa de Sistemas de Informação, Outubro de 2013, Universidade de Évora, em Évora, EU

7.5.3. Artigos em Jornais

- Fórum Armazenamento, Semana Informática 591, pag. 11, 8 de Março de 2002
- Empresas Optimizam Informação – ILM, Semana Informática 748, pag.10, 24 de Junho de 2005
- Benefícios de uma gestão virtual dos sistemas de apoio ao negócio, Semanário Económico, Pag 17, 16-22 Março 2007
- Quando eles ficam em casa (teletrabalho), Pag 106, Revista Exame nº 267, Julho 2006

7.5.4. Participações em seminários como speaker

- Évora University – Last year students Lecture about technology IBM Conferences (BP and Clients) – *et al.*

7.5.5. Submissão Papers, publicações e projetos de pesquisa

Investigador na ADETTI

7.6. Informações Complementares

7.6.1. Conhecimentos de Francês e Inglês, escrito e falado.

Língua Mãe(s) Português

Outras linguas(s) Inglês e Francês

		Compreensão	Conversação
Nível Europeu (*)	Compreensão	Leitura	Interacção
Língua Inglesa	C1 Expedito	B2 Independente	B1 Independente
Língua Francesa	B2 Independente	B2 Independente	A2 Elementar

Tabela 2 - Nível do Quadro Europeu Comum de Referência (CECR)

7.6.2. Actividades Associativas:

- Exerceu, no ano lectivo de 90/91 o cargo de Presidente do conselho Fiscal do Centro Desportivo Universitário de Évora (CDUE).
- No mesmo ano foi membro directivo da Associação de Estudantes da Universidade de Évora (AEUE).
- Presidente do Conselho Distrital de Arbitragem (CAD) da Associação de Basquetebol de Évora (1992-1998).
- Membro da Comissão Instaladora do Núcleo do Alentejo da Associação Nacional de Jovens Empresários (1995).
- Exerce as funções de Vice-Presidente da Direcção do Aero Club de Portugal, desde 2012.

7.6.3. Actividades Desportivas:

- Exerceu actividades de gestão de Desporto Escolar e projectos de Férias Desportivas desde 85 até 89.
- Exerce funções na Associação de Basquetebol de Évora (ABE) como Presidente do Conselho de Arbitragem Regional (CAR).

- Exerceu funções de árbitro nacional de Basquetebol.
- Fez parte do Corpo de Prelectores da Escola Nacional de Basquetebol.
- Desempenha funções em pequenos polos de desenvolvimento da modalidade de Basquetebol
- Regularmente praticava desportos de ar livre tais como montanhismo, “rafting”, “hydrospeed” e “canyoning”

7.6.4. Outras Informações:

- Possui carta de Marinheiro
- Piloto Privado de Aviação Geral
- Carta de Condução de moto e de ligeiros
- Ordem dos Engenheiros: 51919

Referências bibliográficas

- Alfieri, Roberto; Cecchini, Roberto; Ciaschini, Vincenzo; dell'Agnello, Luca; Gianoli, Alberto; Spataro, Fabio; Bonnassieux, Franck; Broadfoot, Philippa J.; Lowe, Gavin; Cornwall, Linda; Jensen, Jens; Kelsey, David P.; Frohner, Ákos; Groep, David L.; Cerff, Wim Som de; Steenbakkers, Martijn; Venekamp, Gerben; Kouril, Daniel; McNab, Andrew; Mulmo, Olle; Silander, Mika; Hahkala, Joni; Lörentey, Károly - Managing Dynamic User Communities in a Grid of Autonomous Resources. *CoRR*. Vol. cs.DC/0306. (2003).
- Anderson, D. P. - BOINC: a system for public-resource computing and storage. In: Grid Computing, 2004. Proceedings. Fifth IEEE/ACM International Workshop on, 2004, 8 Nov. 2004.
- Arvidson, W. - Forming an attachment. Network-attached storage for electronic document management. *MGMA Connex*. Vol. 3. n.º 6 (2003). pp. 28-9.
Disponível em WWW: <<http://www.ncbi.nlm.nih.gov/pubmed/12886717>>.
1537-0240 (Print) 1537-0240 (Linking)
- Blanchet, C.; Mollon, R.; Deleage, G. - Building an encrypted file system on the EGEE grid: application to protein sequence analysis. In: Availability, Reliability and Security, 2006. ARES 2006. The First International Conference on, 2006, 20-22 April 2006.
- Foster, I.; Kesselman, C. - The Globus Project: A Status Report. In: Washington, {DC}, {USA}, 1998, 1998. <<http://portal.acm.org/citation.cfm?id=795689.797877>>
- Foster, Ian; Kesselman, Carl - The grid : blueprint for a new computing infrastructure. San Francisco: Morgan Kaufmann Publishers, 1999. 1558604758
9781558604759
- Foster, Ian; Kesselman, Carl; Nick, Jeffrey M.; Tuecke, Steven - The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration. In: 2002, 2002.
- Fuller, R. Buckminster; McHale, John - World design science decade, 1965-1975:: Five two-year phases of a world retooling design proposed to the International Union of Architects for adoption by world architectural schools. Southern Illinois

- University, 1963. Disponível em WWW: <<http://www.amazon.co.uk/World-design-science-decade-1965-1975/dp/B0006BMGIW>>.
- Garfinkel, Simson - PGP: Pretty Good Privacy. {O'Reilly Media}, 1994. Disponível em WWW: <<http://www.amazon.ca/exec/obidos/redirect?tag=citeulike09-20&path=ASIN/1565920988>>. 1565920988
- Globus - Globus Toolkit. 2013. Disponível em WWW: <<http://toolkit.globus.org/toolkit/>>.
- Hevner, Alan R.; March, Salvatore T.; Park, Jinsoo; Ram, Sudha - Design science in information systems research. *MIS Quarterly*. Vol. 28. n.º 1 (2004). pp. 75-105. Disponível em WWW: <<http://dl.acm.org/citation.cfm?id=2017212.2017217>>.
- Information, Federal - Announcing the ADVANCED ENCRYPTION STANDARD (AES) 2001. Disponível em WWW: <<http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf>>.
- Johansson, Jesper M. - On the impact of network latency on distributed systems design. *Inf. Technol. and Management*. Vol. 1. n.º 3 (2000). pp. 183-194. Disponível em WWW: <<http://dl.acm.org/citation.cfm?id=595252.595281>>. 1385-951X
- Junrang, Li; Zhaohui, Wu; Jianhua, Yang; Mingwang, Xia - A secure model for network-attached storage on the grid. In: *Services Computing, 2004. (SCC 2004). Proceedings. 2004 IEEE International Conference on, 2004, 15-18 Sept. 2004*.
- Kerckhoffs, Auguste - La cryptographie militaire. *Journal des sciences militaires*. Vol. IX. (1883). pp. 5-83. Disponível em WWW: <<http://www.petitcolas.net/fabien/kerckhoffs/>>.
- Leach, P.; Mealling, M.; Salz, R. - A universally unique identifier (UUID) URN namespace. 2005. Disponível em WWW: <<http://www.ietf.org/rfc/rfc4122.txt>>.
- Montagnat, J.; Jouvenot, D.; Pera, C.; Frohner, A.; Kunszt, P. Z.; Koblitz, B.; Loomis, C. - Implementation of a Medical Data Manager on top of gLite services. 2006. Disponível em WWW: <<http://cds.cern.ch/record/%20941801/files/egee-tr-2006-002.pdf>>.

- Rivest, R. L.; Shamir, A.; Adleman, L. - A method for obtaining digital signatures and public-key cryptosystems. *Commun. ACM*. Vol. 21. n.º 2 (1978). pp. 120-126.
Disponível em WWW: <<http://dx.doi.org/10.1145/359340.359342>>. 0001-0782
- Seitz, L.; Pierson, J.; Brunie, L. - Key Management for Encrypted Data Storage in Distributed Systems. In: Security in Storage Workshop, 2003. SISW '03. Proceedings of the Second IEEE International, 2003, 31-31 Oct. 2003.
- Sim, Alex; Berkeley, Lawrence - Grid , Storage and SRM. (2008). 2146782625
- Sim, Alex; Gu, Junmin; Shoshani, Arie - Berkeley - SRM v2.1.1. (2005). pp. 1-35.
- Simon, H. - The Sciences of the Artificial. 3. MIT Press, 1996. Disponível em WWW: <<http://m.friendfeed-media.com/092e5a73c91e0838eeb11e0fe90edaf9e9afc065>>. 0262691914
- Simon, Herbert - Sciences of the Artificial. MIT Press, 1967. Disponível em WWW: <<http://www.amazon.co.uk/Sciences-Artificial-Herbert-Simon/dp/0262691914>>. 0262691914 ISBN-13: 978-0262691918 Product Dimensions: 15.2 x 1.5 x 22.9 cm Average Customer Review: <SCRIPT type=“text/javascript”> function reviewHistPingAjax() { jQuery.get("/gp/customer-reviews/common/du/recordHistoPopAj
- Sordi, José Osvaldo De; Meireles, Manuel; Sanches, Cida - DESIGN SCIENCE APLICADA ÀS PESQUISAS EM ADMINISTRAÇÃO: REFLEXÕES A PARTIR DO RECENTE HISTÓRICO DE PUBLICAÇÕES INTERNACIONAIS. *Revista de Administração e Inovação*. Vol. 8. n.º 1 (2011). pp. 10-36. Disponível em WWW: <http://www.revistarai.org/rai/article/viewFile/770/pdf_22>.
- Thain, D.; Livny, M. - Parrot: Transparent User-Level Middleware for Data-Intensive Computing. 2005. Disponível em WWW: <<http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.9.8435>>.
- Walls, J. G.; Widmeyer, G. R.; El Sawy, O. A. - Building an Information System Design Theory for Vigilant EIS. *Information Systems Research*. Vol. 3. n.º 1 (1992). pp. 36-59. Disponível em WWW: <<http://isr.journal.informs.org/content/3/1/36.citation>>.

Zeng, Wenying; Zhao, Yuelong; Ou, Kairi; Song, Wei - Research on cloud storage architecture and key technologies. In: ICIS '09, New York, New York, USA, 2009, <<http://dl.acm.org/citation.cfm?doid=1655925.1656114>>