



Licenciatura em Gestão de Sistemas e Computação

Gestão de Identidades em ambientes UNIX

Projecto Final de Licenciatura

Elaborado por Ana Paula Joaquim Gonçalves

Aluno nº 19990480

Orientador: Professor Dr. Sérgio Nunes

Barcarena

Novembro 2011

Universidade Atlântica

Licenciatura em Gestão de Sistemas e Computação

Gestão de Identidades em ambientes UNIX

Projecto Final de Licenciatura

Elaborado por Ana Paula Joaquim Gonçalves

Aluno nº 19990480

Orientador: Professor Dr. Sérgio Nunes

Barcarena

Novembro 2011

O autor é o único responsável pelas ideias expressas neste relatório

Agradecimentos

Ao Professor Dr. Sérgio Nunes pela orientação e disponibilidade ao longo da elaboração e evolução do projecto.

Aos colegas que se mostraram disponíveis e que pelo seu apoio e troca de ideias, contribuíram para a sua elaboração.

Aos amigos pelos estímulos e pelas suas palavras de incentivo.

E principalmente à família que sempre me incentivou e ajudou para a realização e concretização deste projecto.

Resumo

Gestão de Identidades em ambientes UNIX

A gestão de utilizadores é um elemento chave na implementação de sistemas seguros. A gestão ineficaz de utilizadores, ou dos seus privilégios permitem que os sistemas fiquem vulneráveis e expostos a ataques internos ou externos. Sendo assim, é importante proteger o acesso aos servidores usando técnicas simples e eficazes na gestão de utilizadores. Consoante o grau de complexidade de segurança implementado nos sistemas aliado à quantidade de palavras-chave (*passwords*) que os utilizadores necessitam de memorizar poderá existir transtornos com elevados custos operacionais, como por exemplo, na formação de uma equipa maior de suporte ao utilizador (*help desk*) ou no reforço da equipa de administração dos sistemas.

Este trabalho tem como objectivo apresentar e desenvolver uma infra-estrutura flexível e dinâmica de apoio à criação, manutenção e gestão de identidades com o objectivo de melhorar o acesso dos utilizadores às aplicações, facilitar a administração de sistemas, minimizar a complexidade e reduzir os custos de exploração desta área nas organizações.

Palavras-chave: Gestão de Identidades, Acesso, Utilizadores, Sistemas UNIX, Segurança.

Abstract

Identity Management for UNIX

User management is a key element in the implementation of secure systems. The ineffective user management and, or its privileges allow systems to become vulnerable and exposed to internal and external attacks. Thus is important to protect access to servers using simple and effective techniques in user management. According to the degree of security complexity implemented in systems together with amount of passwords users need to memorize there may be disturbances with high operational costs, such as in forming a larger help desk team or reinforcement of the system management team.

This work aims to present and develop a flexible and dynamic infrastructure to support the creation, maintenance and management of identities with the goal of improving user access to applications, facilitate system administration, and minimize complexity and reducing operating costs from this area in organizations.

Keywords: Identity Management, Access, Users, UNIX Systems, Security.

Índice

Agradecimentos.....	iii
Resumo.....	iv
Abstract	v
Índice.....	vi
Índice de figuras.....	viii
Índice de tabelas.....	ix
Lista de abreviaturas e siglas.....	x
Introdução	1
1. Sistema Operativo UNIX.....	3
1.1. Problema.....	3
1.2. Sistemas UNIX.....	4
1.3. Estrutura de Objectos/Ficheiros.....	5
1.4. Estrutura de directórios.....	7
1.5. Utilizadores.....	8
1.6. Ficheiro /etc/passwd e /etc/group	9
1.7. Permissões	13
1.8. Acesso.....	15
2. Gestão de Identidades	17
2.1. Identidade.....	17
2.2. Autenticação	18
2.3. Autorização.....	19
2.4. Gestão de acessos.....	20
2.5. Sistema de Gestão de Identidades.....	21

2.6. Produtos <i>IdM</i>	22
2.7. Sucesso Vs Insucesso dos produtos <i>IdM</i>	31
3. Gestão de Identidades em Ambientes UNIX	34
3.1. Gestão de Utilizadores nos sistemas UNIX.....	34
3.2. Solução SI Gestão de Identidades.....	35
Conclusão	63
Bibliografia	66
Anexos.....	69
Portal SI Gestão de Identidades	69

Índice de figuras

Fig. 1 - Camadas Básicas do UNIX	5
Fig. 2 - Estrutura de directórios.....	7
Fig. 3 - Estrutura do ficheiro <i>/etc/passwd</i>	9
Fig. 4 - Estrutura do ficheiro <i>/etc/groups</i>	11
Fig. 5 – Informação UNIX	14
Fig. 6 - Permissões UNIX	14
Fig. 7 – Conceito de Identidade	18
Fig. 8 – Ciclo de vida de uma Identidade.....	20
Fig. 9 – <i>Microsoft Forefront Identity Manager (FIM)</i>	22
Fig. 10 – <i>Microsoft Forefront Identity Manager (FIM)</i> - áreas de actuação	24
Fig. 11 – <i>Oracle Identity Manager (OIM)</i> – <i>Consola</i>	26
Fig. 12 – <i>Tivoli Identity Manager</i>	30
Fig. 13 – Diagrama Caso de Utilização: S.I. Gestão de Identidades	38
Fig. 14 – Diagrama de classes: S.I. Gestão de Identidades.....	41
Fig. 15 – Diagrama de Objectos: Solicitar Acesso	44
Fig. 16 – Diagrama de sequência: Validar Acesso	45
Fig. 17 – Diagrama de sequência: Solicitar Acesso / Alterar Perfil.....	46
Fig. 18 – Diagrama de sequência: Desbloquear / Activar Acesso e Eliminar Acesso	47
Fig. 19 – Diagrama de sequência: Consultar Acessos / Perfil, Consultar Pedidos e Revalidar Acessos	48
Fig. 20 – Diagrama de estados: Validar Acesso	50
Fig. 21 – Diagrama de estados: Seleccionar Formulário	51
Fig. 22 – Diagrama de actividades: Validar Acesso	52

Fig. 23 – Diagrama de actividades: Solicitar Acesso / Alterar Perfil	54
Fig. 24 – Diagrama de actividades: Desbloquear / Activar Acesso e Eliminar Acesso..	56
Fig. 25 – Diagrama de actividades: Consultar Acessos / Perfil, Consultar Pedidos e Revalidar Acessos	57
Fig. 26 – Diagrama de componentes: S.I. Gestão de Identidades.....	58
Fig. 27 – Diagrama de instalação: S.I. Gestão de Identidades.....	60
Fig. 28 – S.I. Gestão de Identidades: Página Login.....	69
Fig. 29 – S.I. Gestão de Identidades: Página Menu Principal.....	70
Fig. 30 – S.I. Gestão de Identidades: Página Consultar Acessos / Perfil.....	71
Fig. 31 – S.I. Gestão de Identidades: Página Solicitar Acesso	72
Fig. 32 – S.I. Gestão de Identidades: campo informativo.....	73
Fig. 33 – S.I. Gestão de Identidades: Página Alterar Perfil	74
Fig. 34 – S.I. Gestão de Identidades: Página Desbloquear / Activar Acesso.....	75
Fig. 35 – S.I. Gestão de Identidades: Página Consultar Estado do Pedido	76
Fig. 36 – S.I. Gestão de Identidades: Página Autorizar Pedido	77
Fig. 37 – S.I. Gestão de Identidades: Página Revalidar Acessos.....	78
Fig. 38 – S.I. Gestão de Identidades: Página Eliminar Acessos	79

Índice de tabelas

Tabela 1 – Directorias UNIX.....	7
Tabela 2 – Ficheiro <i>passwd</i> e <i>group</i>	9
Tabela 3 – Campos do ficheiro <i>/etc/passwd</i>	10
Tabela 4 – Campos do ficheiro <i>/etc/group</i>	11
Tabela 5 – Tipos de acesso.....	14

Lista de abreviaturas e siglas

AD – Active Directory

API - Application Programming Interface

ASCII - American Standard Code for Information Interchange

FIM 2010 - Microsoft Forefront Identity Manager 2010

HIPAA - Health Insurance Portability and Accountability Act of 1996

HTTP - Hypertext Transfer Protocol

HTTPS - HyperText Transfer Protocol Secure

IAM - Identity and Access Management

IEC - International Electrotechnical Commission

ISO - International Organization for Standardization

ITIL - Information Technology Infrastructure Library

JAVA EE - Java Platform, Enterprise Edition

OASIS - Organization for the Advancement of Structured Information Standards

OIM - Oracle Identity Manager

OMG - Object Management Group

PC – Personal Computer

PIN - Personal identification number

SI – Sistemas de Informação

SSH - Secure Shell

SSL - Secure Sockets Layer

SOX - Sarbanes-Oxley Act

TCB - Trusted Computing Base

TCP/IP - Transmission Control Protocol/Internet Protocol

TI – Tecnologias de Informação

UML - Unified Modeling Language

VPN - Virtual Private Network

Introdução

A nossa sociedade está cada vez mais dependente de infra-estruturas informáticas e, em especial, dos sistemas de informação.

A constante evolução dos mercados afectam o funcionamento das organizações. O rápido avanço tecnológico e a crescente pressão da concorrência, fazem com que as organizações necessitem de recorrer com maior frequência às tecnologias e a sistemas de informação para suporte do seu *core business*, esteja ele relacionado ou não, com as tecnologias e sistemas de informação.

A segurança deve ser uma preocupação desde o início da concepção de qualquer tecnologia ou sistema de informação, pelo que deve ser implementada através de políticas de segurança e suportadas pela gestão.

A necessidade de adequação a regulamentações como o *Sarbanes-Oxley Act* (SOX) [SOX, 2006] em conjunto com os diferentes modelos de gestão, dos quais podemos considerar os modelos: *CobiT* [COBIT, 2011] para a gestão de TI; o *ITIL* [ITIL, 2007] para a gestão de serviços de TI; e o *ISO/IEC 27002:2005* (antigo *ISO/IEC 17799:2005*) [ISO, 2011] para a gestão de segurança da informação, entre outros, asseguram a conformidade com as melhores práticas de processos e segurança da informação, no entanto, podem levar à implementação de políticas de segurança com alguma complexidade para os utilizadores finais. A adequação a esses padrões internacionais em conjunto com a gestão individual que é requerida, implicam custos para as organizações e podem significar perda de competitividade no mercado a curto prazo.

Considerando a problemática acima, este trabalho pretende analisar em pormenor a questão da gestão de identidades nas organizações, os produtos existentes no mercado, as suas vantagens e desvantagens. Pretende igualmente apresentar uma solução para a gestão de identidades customizada e simplificada tendo em conta o utilizador final, que geralmente está pouco susceptível aos assuntos de segurança nas organizações.

Sendo assim, o trabalho foi estruturado em três capítulos, sendo que no primeiro capítulo foi elaborado uma breve introdução sobre a estrutura e funcionamento dos

sistemas UNIX independentemente da plataforma, com o intuito do leitor familiarizar-se com os respectivos conceitos e com a questão da segurança nestes sistemas.

O segundo capítulo pretende enquadrar os sistemas/soluções de Gestão de Identidades. Foram analisados alguns dos produtos existentes no mercado, referindo-se as suas vantagens e benefícios. Ainda neste capítulo, e a título de conclusão foi apresentado o motivo que levou ao desenvolvimento do solução desenvolvida no capítulo seguinte.

Sendo assim, no capítulo três foi elaborado novo enquadramento dos sistemas UNIX, enquanto sistemas que podem ser normalizados e regulados com as directrizes de segurança. Foi elaborada a análise de requisitos funcionais para a implementação de uma solução de Gestão de Identidades vocacionada sobretudo para os fluxos de autorizações e no relacionamento com os utilizadores, com recurso à linguagem UML, descrevendo-se os casos de utilização e os diferentes diagramas no contexto do projecto e no âmbito da solução.

Por último foram elaboradas as conclusões sobre o projecto e sobre o trabalho em si.

1. Sistema Operativo UNIX

1.1. Problema

A gestão de contas de utilizadores em ambientes UNIX®¹ torna-se difícil de gerir na medida em que aumenta o número de servidores e de utilizadores. Essa situação agrava-se quando os sistemas em questão necessitam de obedecer a determinados padrões de conformidade e a controlos de auditoria exigidos por uma política de segurança.

Uma política de segurança deve ser definida e, citando Henrique Mamede [Mamede, 2006, p.37] “tomando em consideração todo um conjunto de factores externos e internos à própria organização.”. Como factor externo, Mamede considera o meio envolvente: a legislação, custos e riscos, e identifica como factores internos, por exemplo os objectivos da segurança, os custos com a implementação, os sistemas computacionais envolvidos, entre outros.

É necessário igualmente diferenciar segurança de protecção. Segundo Alberto Carneiro [Carneiro, 2009, p.36] enquanto as políticas de segurança são linhas orientadoras na protecção dos dados e de recursos, os mecanismos de protecção são instrumentos de operacionalização das políticas. Os sistemas operativos possuem mecanismos próprios de protecção que permitem aplicar algumas das recomendações definidas nas políticas de segurança da organização.

O objectivo da segurança informática ou segurança em sistemas informáticos é proteger um conjunto de bens (equipamentos e dados) no sentido de preservar o valor que possuem para um indivíduo ou para a organização, assegurar a continuidade do negócio, minimizar possíveis danos e maximizar o retorno dos investimentos. Na definição da segurança informática devem ser considerados três princípios básicos:

- Confidencialidade – no sentido em que a informação só deve estar acessível para quem tem acesso autorizado (acesso ao estritamente necessário);

- Integridade – relacionado com a credibilidade e a exactidão da informação, assim como o(s) método(s) de processamento (protecção da respectiva fiabilidade e origem).
- Disponibilidade – relacionado com o acesso e utilização da informação sempre que é solicitada por um utilizador autorizado (garantia do acesso autorizado sempre e na medida do necessário).

É necessário conhecer os sistemas UNIX nomeadamente a sua estrutura, o seu modo de funcionamento e a interacção com os utilizadores e adequar os mecanismos de protecção para dar resposta às linhas orientadoras descritas na política de segurança, garantindo os princípios básicos da segurança da informação.

1.2. Sistemas UNIX

Os sistemas Unix caracterizam-se por sistemas operativos multi-tarefa (*multitasking*) em que é possível repartir a utilização do processador (*CPU*) por diversas tarefas ou programas em simultâneo; e por sistemas operativos multi-utilizador (*multi-user*), onde é permitido o acesso concorrente de vários utilizadores ao sistema.

Inicialmente foi escrito em linguagem *assembly* [*Assembly language*, 2011] (linguagem de baixo nível), e mais tarde reescrito em C [*C (programming language)*, 2011], atribuindo-lhe assim outra característica - a de portabilidade, ou seja a possibilidade de ser executado (ou facilmente recompilado) em outras plataformas para além daquela que esteve na sua origem.

O sistema operativo UNIX foi concebido em diversas camadas (Figura 1), que formam uma hierarquia através das quais os utilizadores utilizam os recursos de processamento da máquina. Na camada mais interna está o *hardware*, composto de componentes físicos. A camada seguinte e, que envolve o *hardware* é o *kernel* e é responsável pela gestão e controlo do *hardware* e do sistema em si. O *kernel*, por sua vez, está envolto por programas (comandos) que realizam tarefas específicas. A camada mais externa é a *shell* e é responsável pela interacção entre o utilizador e o sistema operativo.

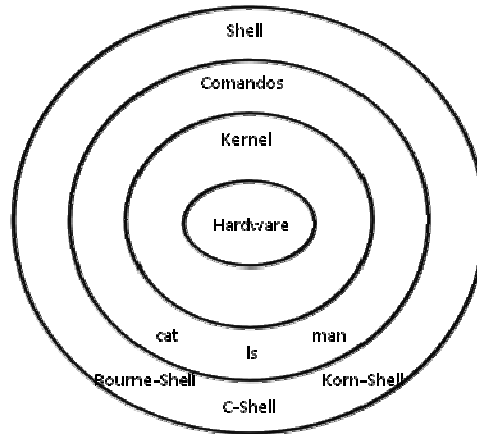


Fig. 1 - Camadas Básicas do UNIX

1.3. Estrutura de Objectos/Ficheiros

O UNIX é composto por diversos tipos de objectos, designados genericamente por ficheiros. Os mais comuns consistem em:

- Ficheiros regulares (*Ordinary Files*): São ficheiros normais que contêm diversos dados, que podem ser informações em texto *ascii*, como dados de aplicações e/ou programas. São representados pelo “-” (hífen).

```
-rw-r--r-- ... /etc/passwd
```

- Directorias (*Directories*): São ficheiros do tipo directorias. Podem conter ficheiros regulares, ficheiros especiais ou outras directorias (subdirectorias). São usadas essencialmente para organizar os grupos de ficheiros e nunca contêm dados (texto). Todos os ficheiros dependem da directoria *root* (*root directory*). Os ficheiros do tipo directorias são representados pela letra “d”.

```
drwxr-xr- ... /dev
```

- Ficheiros que representam um *link* simbólico (*Symbolic link*): são ficheiros que são uma referência a outros ficheiros, isto é, que são um atalho para outros ficheiros. São representados pela letra “l” (L em minúsculas).

```
lrwxrwxrwx ... /tmp->/var/tmp
```

d) Ficheiros Especiais: Designação atribuída aos ficheiros utilizados pelo *kernel* em rotinas que providenciam acesso a algumas funcionalidades do sistema operativo. São usados para representar um dispositivo físico real como impressoras, discos, terminais e para as operações de *Input/Output (I/O)*. Geralmente encontram-se por baixo da directoria “/dev” e podem ser de dois tipos:

- a. Ficheiros especiais de caracteres (*character special files*) – relacionados com dispositivos cujos dados são transmitidos carácter a carácter. O sistema lê ou escreve imediatamente a partir do dispositivo sem ser necessário recorrer à memória intermédia (*buffering*). São representados pela letra “c” (exemplo de terminais virtuais, teclados e *modems*).

```
crw-rw-rw- ... tty00
```

- b. Ficheiros especiais de bloco (*block special files*) – correspondem a dispositivos que transmitem os dados em forma de blocos. O sistema lê ou escreve nestes dispositivos em blocos múltiplos geralmente de 512 ou 1024 *bytes*. (exemplo dos discos físicos, *CD-ROMs*, e determinadas regiões de memória).

```
brw----- ... /dev/disk/cdrom0c
```

e) Ficheiro *Local socket*: ficheiros especiais usados na comunicação inter-processos. Permitem a comunicação entre dois processos. No envio dos dados, os processos podem enviar descritores através da comunicação *socket* usando as funcionalidades de chamadas ao sistema do *sendmsg()* e *recvmsg()*. São representados pela letra “s”.

```
srwxrwxrwx ... /tmp/.X11-unix/X0
```

1.4. Estrutura de directórios

O sistema UNIX é baseado numa estrutura organizada e hierárquica de directorias que contém outras directorias e/ou ficheiros. A directoria de topo é chamada directoria raiz ou *root directory*, representada pela “ / ” (barra). Por baixo desta, existem outras directorias que poderão dar origem a subdirectorias, criando assim as ramificações de uma árvore invertida.

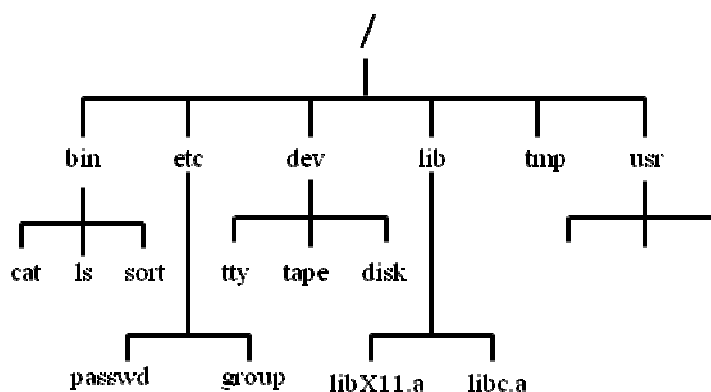


Fig. 2 - Estrutura de directórios

Cada uma das directorias tem uma função específica, como pode ser observado na tabela abaixo:

Diretoria	Descrição
/	Directoria principal (raiz) de um sistema UNIX
/bin	Contêm os comandos e/ou programas <i>standard</i> do sistema UNIX.
/etc	Contêm os comandos e/ou programas de administração e configuração do sistema UNIX.
/dev	Contêm os comandos e /ou programas que permitem controlar os dispositivos (periféricos) dos sistemas UNIX, tais como terminais, discos, medias (<i>tapes</i>), entre outros.
/lib	Contêm livrarias usadas pelos comandos e/ou programas do sistema UNIX.
/tmp	Usada para guardar ficheiros temporários que estão a ser acedidos ou que deixam de ser precisos depois de terminar um comando e/ou programa.
/usr	Contêm outras subdirectorias como por exemplo, referentes ao <i>scheduling</i> das impressoras, comandos, programas extras e/ou livrarias específicas. Em alguns sistemas operativos UNIX contêm igualmente as directoria dos utilizadores.

Tabela 1 – Directorias UNIX

1.5. Utilizadores

Nos sistemas UNIX existem três tipos de contas de utilizadores:

Utilizador *root* ou Administrador - Utilizador privilegiado com acesso a todos os ficheiros, independentemente de quem é o proprietário ou as respectivas permissões. Este utilizador é responsável por administrar e manter todo o sistema. Normalmente é conhecido por *root*. A sua *homedir* é a directoria principal do sistema (*root directory*), da qual dependem todas as outras como referido no subcapítulo anterior.

Contas de sistema – São contas específicas para determinadas funções de sistema, como por exemplo, conta de *ssh* ou *sshd* para a funcionalidade *ssh*, conta *lp* ou *lpd* relacionada com funcionalidade das impressoras, etc. As contas do sistema são geralmente fornecidas pelo sistema operativo durante a sua instalação ou aquando da instalação de um software. Por serem necessárias a determinadas funções de sistema não devem ser alteradas a não ser que se pretenda desactivar o tipo de funcionalidade tendo em consideração as implicações dessa modificação. Sendo contas de funções de sistema geralmente não permitem que seja efectuado acesso directo com elas.

Utilizadores Normais - São os utilizadores com direitos limitados. Possuem a sua *homedir* na directoria ou subdirectoria dedicada aos utilizadores (exemplo: */home/<utilizador>* ou em */usr/users/<utilizador>*). Após autenticação no sistema, os utilizadores são automaticamente posicionados na respectiva directoria de trabalho.

O modo de actuação nos sistemas UNIX é baseado em alguns mecanismos, nomeadamente:

- num sistema de autenticação para identificação de cada utilizador.
- num sistema de objectos com permissões e propriedades sobre os ficheiros.

1.6. Ficheiro `/etc/passwd` e `/etc/group`

A informação da conta de um utilizador fica guardada em ficheiros como o `/etc/passwd` e no `/etc/group`, transversal a todos os sistemas UNIX. Dependendo do nível de segurança instalado, nomeadamente com a passagem a *trusted systems* ou para o nível C2, poderão existir outros ficheiros e/ou base de dados na estrutura UNIX (exemplos na tabela 2) que contêm igualmente informação sobre as contas dos utilizadores.

User Accounts		AIX	HP-UX	LINUX	Solaris	Tru64
Password files	default	<code>/etc/passwd</code>	<code>/etc/passwd</code>	<code>/etc/passwd</code>	<code>/etc/passwd</code>	<code>/etc/passwd</code>
	trusted systems	<code>/etc/security/passwd</code>	TCB files	<code>/etc/shadow</code>	<code>/etc/shadow</code>	TCB files
Group files	default	<code>/etc/group</code>	<code>etc/group</code>	<code>/etc/group</code>	<code>/etc/group</code>	<code>/etc/group</code>
	trusted systems	<code>/etc/security/group</code>				

Tabela 2 – Ficheiro `passwd` e `group`

Independentemente dos sistemas estarem em *trusted system*, os ficheiros `/etc/passwd` e `/etc/groups` tem de existir nos sistemas.

O ficheiro `/etc/passwd` é um simples ficheiro de texto (formato *ascii*), que contém uma linha que define os atributos básicos de cada utilizador. Dentro de cada linha, existem vários campos de informação, separados por “:” (dois pontos), como podemos ver na figura 3.

```
<username>:*:<uid>:<gid>:<gecos>:<homedir>:<shell>
```

Fig. 3 - Estrutura do ficheiro `/etc/passwd`

Cada campo corresponde a determinada informação, com as seguintes características:

Coluna	Campo	Definição
1	<i>Username</i> <i>Login Name</i>	Nome que identifica o utilizador no sistema. Deve ser único. Dependendo dos sistemas operativos existem restrições deste campo, por exemplo: <ul style="list-style-type: none"> - não pode conter espaços, - não devem ser utilizados caracteres especiais, tais como, \$@/[]; =,*?(){}'"`#,\ - em alguns sistemas deve começar por um letra, - pode existir um número limite de caracteres (geralmente 8 caracteres).
2	<i>Encrypted password</i>	Campo da palavra-chave (<i>password</i>). Nos sistemas com um nível de segurança implementado este campo pode estar representado por um * (asterisco) (HPUX e TRU64), por um ! (ponto de exclamação) (AIX) ou pela letra x (SOLARIS e LINUX). Caso o sistema não esteja em <i>trusted system</i> , a <i>password</i> fica neste campo e apesar de estar encriptada, pode ser visualizada por qualquer utilizador que se autentique no sistema e consulte o ficheiro.
3	<i>UserID</i> <i>UID</i>	(<i>User Identifier</i>). Número atribuído a um utilizador. Deve ser único no sistema. Usado nas operações internas pós-logins, sobretudo para definir permissões e dessa forma controlar o acesso aos dados. Tal como no campo do <i>username</i> existem algumas regras e convenções: <ul style="list-style-type: none"> - campo com um valor numérico decimal. - o valor 0 (zero) é atribuído ao <i>superuser (root)</i>. - o valores entre 1-100 são reservados aos utilizadores de sistema (excepto o utilizador <i>nobody</i> que recebe o valor contrário ao utilizador <i>root</i>, p.ex. uid 32767 ou dentro do <i>range</i> 65530–65535 (em HPUX que recebe o valor -1).
4	<i>GID</i>	(<i>Group Identifier</i>). Grupo principal a que um utilizador pertence. Valor numérico decimal único com correspondência no ficheiro <i>/etc/group</i> . Usado para definir permissões e dessa forma controlar o acesso aos dados.
5	<i>Gecos</i>	(<i>General Comprehensive Operating System</i>). Campo “comentário” reservado para o preenchimento de informação relativa ao utilizador como por exemplo, nome completo, identificação do local de trabalho (edifício e, ou sala) telefone de serviço. O seu valor é texto, limitado a 100 caracteres e não pode conter : (dois pontos).
6	<i>HomeDir</i>	(<i>Home Directory</i>). Localização da directoria de trabalho do utilizador. Habitualmente é atribuído a designação de <i>HOME</i> e dependendo do sistema operativo tem uma localização <i>default</i> . (<i>PathDefault=\$HOME/<username></i>)
7	<i>Shell</i>	Programa inicial invocado quando é estabelecida uma sessão no sistema. Interface ou interpretador de comandos entre o utilizador e o sistema. Os mais comuns são: <i>Bourne shell (sh)</i> , <i>C shell (csh)</i> , <i>Korn-shell (ksh)</i> e <i>Bourne-Again shell (bash)</i> .

Tabela 3 – Campos do ficheiro */etc/passwd*

O ficheiro */etc/group* é igualmente um ficheiro simples de texto onde são referidos os grupos do sistema. À semelhança do que acontece no ficheiro */etc/passwd*, cada linha corresponde a um grupo. Os grupos permitem juntar vários utilizadores de acordo com a função comum que desempenham. O ficheiro */etc/groups* tem a(s) seguinte(s) estrutura(s):

```
<groupname>:*:<gid>:
```

Ou

```
<groupname>:*:<gid>:<username>
```

Fig. 4 - Estrutura do ficheiro */etc/groups*

Onde, cada campo tem as seguintes características:

Campo	Tipo de Permissão	Descrição
1	<i>Group name</i>	Define o nome do grupo. Deve ser único. Composição/restrições semelhantes às do campo <i>username</i> do ficheiro <i>/etc/passwd</i> .
2	<i>Encrypted Password</i>	Campo palavra-chave (<i>password</i>). Por norma este campo está representado por um * (asterisco) ou pela letra x, equivalente a sem (<i>null password</i>).
3	<i>GID</i>	(Group Identifier) . Número atribuído ao grupo. Valor numérico decimal único. Usado nas operações internas pós-login e sobretudo para definir direitos sobre os dados.
4	<i>Username</i>	Utilizadores (<i>usernames</i>) que pertencem ao grupo, enquanto grupo secundário. Mencionados e separados pela , (virgula). No caso de ser grupo primário, o <i>username</i> não necessita de ser mencionado, uma vez que a correspondência é directa (4º campo no <i>/etc/passwd</i> → 3º campo deste ficheiro).

Tabela 4 – Campos do ficheiro */etc/group*

Um dos problemas suscitados pelo ficheiro */etc/passwd* é o facto do ficheiro ter de ser legível para todos os utilizadores o que levanta *issues* em relação à segurança, apesar dos sistemas UNIX utilizarem a função criptográfica *salt* que permite adicionar complexidade à palavra-chave que é criada e armazenada.

O mecanismo *salt* consiste na geração de *random bits* (bits aleatórios) que são adicionados à palavra-chave ou frase secreta (*passphrase*). Actualmente esses *bits* nos sistemas UNIX podem ir entre 48 a 128 *bits* (nos sistemas mais antigos estavam limitados a 12-bit) o que aumenta o comprimento e potencialmente a complexidade do *hash* das palavras-chave. Para o utilizador esta situação é transparente e não constitui qualquer alteração à palavra-chave escolhida, no entanto, garante maior segurança uma vez que as *passwords* são salvaguardadas e registadas baseadas na encriptação da palavra-chave + algoritmo *salt*.

Independentemente dessa segurança (palavra-chave encriptada + *salt*) a informação fica registada num ficheiro legível, que com determinadas ferramentas e tempo (entenda-se bastante tempo e espaço para possuir uma base de dados que permita estabelecer “n” comparações e possibilidades), é possível decifrar as mesmas.

Ainda como resposta a esta situação (fragilidade do ficheiro */etc/passwd* ser “público”, isto é, legível por todos) é possível implementar diferentes níveis de segurança nos sistemas UNIX consoante as necessidades e objectivos de segurança. Caso se pretenda um sistema menos seguro, baseado somente na autenticação, opta-se por uma instalação do tipo *base security*, onde a palavra-chave encriptada + *salt* ficam presentes no ficheiro de registo dos utilizadores (sendo que a customização da segurança é também *built-in* do sistema operativo para o nível *base*).

Caso pretenda-se uma solução com um nível de segurança superior pode-se optar por uma solução *trusted*. A solução *trusted* permite proteger dados sensíveis, como é o caso do conteúdo das palavras-chave (*hash* das *passwords* + *salt*), que ficam residentes em ficheiros que somente são legíveis para o utilizador privilegiado, ou seja, o *root*. Por outro lado esta solução também permite adequar e activar outro tipo de políticas de

segurança aos ambientes (como por exemplo, o tempo de vida das *passwords*, a sua reutilização, etc.)

1.7. Permissões

No UNIX utiliza-se os conceitos de propriedade e de permissões que associadas impedem acessos indevidos a ficheiros e/ou às directorias. Para cada objecto UNIX são definidas permissões para três tipos de acesso: leitura, escrita e execução. Estes três tipos de acesso são, por sua vez, atribuídos, a três conjuntos de utilizadores: o proprietário (dono) do objecto, os utilizadores que pertencem ao mesmo grupo do objecto e os restantes.

As regras de permissões no UNIX funcionam da seguinte forma:

- a) Para o utilizador proprietário (*owner/user*) do ficheiro aplicam-se as permissões de *owner*. Representada em UNIX pela letra “u”.
- b) Para o utilizador que não é proprietário do ficheiro, mas pertence ao grupo a que o ficheiro pertence, aplicam-se as permissões do grupo (*group*). Representada em UNIX pela letra “g”.
- c) Para os restantes casos, aplicam-se as permissões dos restantes ou outros (*others*). Representada em UNIX pela letra “o”.

Tal como referido a cada um destes “conjuntos” (*owner*, *group* e *others*) podem ser aplicados os três tipos de acessos: leitura (*read*), escrita (*write*) e execução (*execute*) com o seguinte significado:

Objeto	Tipo de Permissão	Descrição
Ficheiros	Leitura (r)	Permissões que permitem visualizar o conteúdo.
	Escrita (w)	Permissões que permitem alterar o conteúdo ou remover o ficheiro.
	Execução (x)	Permissões que permitem executar o ficheiro.
Diretorias	Leitura (r)	Permissões que permitem listar conteúdo da directoria.
	Escrita (w)	Permissões que permitem criar ou remover ficheiros na directoria.
	Execução (x)	Permissões que permitem mudar para a directoria.

Tabela 5 – Tipos de acesso

A sintaxe que é apresentada quando listamos a informação num sistema UNIX é representada da seguinte forma:

```
-rw-r--r-- <owner> <group> <size> <time> <filename>
```

Fig. 5 – Informação UNIX

Na figura 6, podemos verificar os diferentes conjuntos, permissões e objectos e a forma como os mesmos podem relacionarem-se entre si:

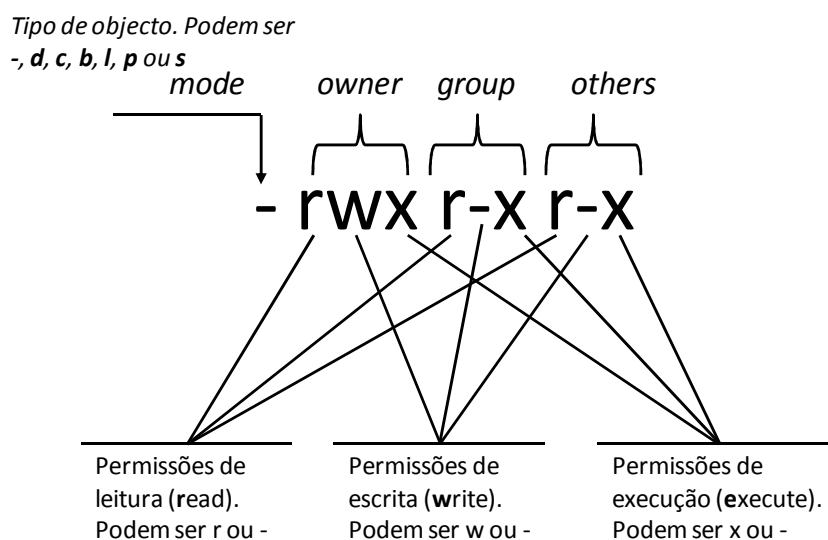


Fig. 6 - Permissões UNIX

O primeiro carácter identifica o tipo de objecto, ou seja, se se trata de um ficheiro regular, directoria ou ficheiro especial. Os nove caracteres seguintes estão divididos em três tipos de acessos. Os três primeiros caracteres mostram as permissões para o proprietário (*owner/user*), que podem ser de leitura (r), escrita (w) ou execução (x). O próximo conjunto de três caracteres mostram as permissões de acesso para os utilizadores do grupo (*groups*), que podem ser igualmente de leitura (r), escrita (w) ou execução (x), e os últimos três caracteres identificam as permissões de acesso para os restantes utilizadores (*others*). O “-” (hífen) significa acesso negado, isto é, que não têm permissão para o tipo de acesso correspondente.

1.8. Acesso

O modo de autenticação nos sistemas UNIX é determinado pela identificação do utilizador e por uma palavra-chave (*password*) associada ao mesmo. No momento da criação do utilizador no sistema é atribuído um nome (*username* ou *loginame*) e uma palavra-chave (*passwd*) ao utilizador, pelo administrador (*root*). Na autenticação ao sistema operativo UNIX feita pelo utilizador, é despoletado o programa *login*, que autentica o utilizador confirmando os dados numa base de dados (normalmente armazenada no ficheiro */etc/passwd* e/ou na estrutura de *trusted* implementada).

Embora o processo de identificação de um indivíduo seja normalmente baseado no *username* e na *password*, em termos de segurança os conceitos de autenticação e de autorização são distintos.

Enquanto que o processo de autenticação tem simplesmente a finalidade de garantir que o indivíduo/utilizador é quem diz ser, o processo de autorização tem como objectivo a atribuição do acesso baseado na identidade do indivíduo.

A autorização de acessos, por sua vez, deve respeitar alguns princípios. Jerome Saltzer e Michael Schroedder [Saltzer, J. H. e Schroedder, M. D., (1974)] enumeram alguns princípios entre os quais o princípio da separação de privilégios e o princípio de

privilégio mínimo. No princípio da separação de privilégios (*Separation of privilege*) pode ser conseguida ao, por exemplo, implementar um segundo mecanismo de validação, em que o utilizador necessita de voltar a colocar credenciais para garantir o acesso para outro nível. Assim, garante-se que acidentes, enganos, ou abuso de confiança são acautelados.

O princípio de privilégio mínimo (*Least privilege*) é o princípio que advoga a atribuição mínima de privilégios necessários (“*need-to-know*”) para a execução das tarefas que estão associadas a um utilizador, para que o mesmo consiga completar a sua actividade correctamente. Desta forma reduz-se o número de potenciais interacções com privilégios superiores aos necessários, minimizando situações de erro e/ou engano.

Enquanto nos sistemas operativos podemos garantir mecanismos de autenticação, é necessário que os mecanismos de autorização sejam igualmente garantidos à priori e ao longo do tempo de vida do acesso. Neste sentido as empresas recorrem a ferramentas e/ou *software* que permitam desenhar sobretudo o processo de autorização referente à concessão dos acessos.

2. Gestão de Identidades

O que surgiu como uma necessidade de gerir as autorizações dos acessos dentro das organizações, rapidamente se tornou na necessidade de gerir os próprios acessos e o tempo de vida de cada um.

Começam a surgir no mercado produtos com o objectivo de gerirem identidades e acessos, que relacionam a gestão de acesso, a gestão das palavras-chave, com identidades, autenticação, autorização e auditoria de segurança.

Mas afinal, no que consiste um sistema de Gestão de Identidades? Em primeiro lugar vamos perceber alguma da terminologia e identificar os requisitos de um sistema de Gestão de Identidades.

2.1. Identidade

Existem várias definições para o termo identidade dentro do contexto de gestão de identidades. Por exemplo, Pfitzmann e Hansen [Pfitzmann, A., e M. Hansen, (2010), p.31], definem identidade como: *“An identity of an individual person may comprise many partial identities of which each represents the person in a specific context or role. A partial identity is a subset of attribute values of a complete identity, where a complete identity is the union of all attribute values of all identities of this person.”* Bishop [Bishop, M., (2002)] entre outros, consideram que o conceito identidade abrange não apenas pessoas, mas também agentes de *software* (como por exemplo: serviços de *Web* e *software* cliente) e dispositivos de *hardware* (como por exemplo, computadores, telemóveis, *PDA*'s, equipamento de rede). Uma identidade pretende definir um conjunto de características essenciais e distintivas de uma pessoa, empresa, sistema, aplicação, ou processo e, que necessitam de ser constantemente actualizados. Pode ser constituída pela combinação de diversos atributos que podem ser permanentes ou temporários, herdados, obtidos ou atribuídos.

Desta forma uma identidade pode ser definida como um conjunto de informações actualizadas, organizadas, codificadas, acessível através de meios técnicos, que

identificam uma pessoa, um objecto ou alguma coisa. A característica principal é que é pública, ou seja, pode ser conhecida por outros (como por exemplo, pessoas, sistemas, ou aplicações).

Na figura abaixo define-se uma relação conceitual possível entre entidades, identidades e atributos.

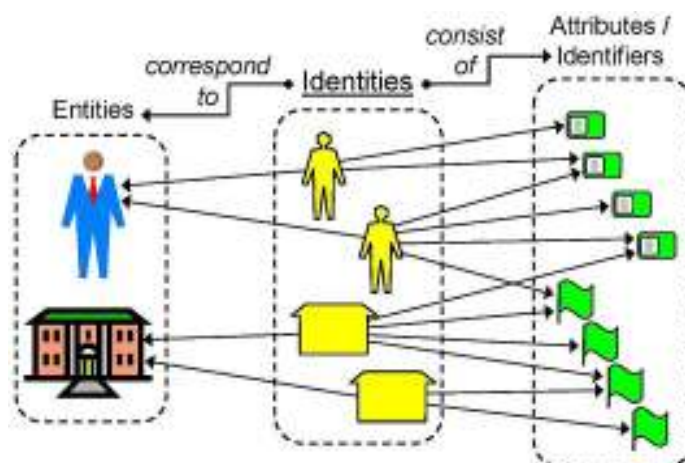


Fig. 7 – Conceito de Identidade

Fonte: http://en.wikipedia.org/wiki/Identity_management

Em traços gerais, uma entidade (que podem ser pessoas, *hardware*, empresas, processos, etc.) pode ter várias identidades, e cada identidade pode ser constituída por diversos atributos (ou características) partilhados ou únicos, estabelecendo-se assim uma relação entre identidades e as entidades que representam, assim como a relação entre as entidades e os atributos que as caracterizam.

2.2. Autenticação

Processo pelo qual é efectuado a confirmação de um atributo, de determinado dado, entidade ou identidade. Resumidamente trata-se da verificação das credencias de uma identidade.

Associado ao controlo de acessos, que pode ser a sistemas, aplicações, e, ou recursos, envolve a confirmação da identidade de uma pessoa ou programa, identificando a sua origem e garantindo que a identidade é quem diz ser.

É privada, ou seja, é (ou pelo menos deve ser) somente do conhecimento da identidade.

Existem diversos métodos de autenticação que permitem verificar a identidade de um utilizador, como por exemplo, palavras-chave (mais comum), *PKI (public-key infrastructure)* certificados digitais, *tokens*, cartões inteligentes, informação biométrica (impressão digital, *scan* da retina), etc. Dependendo do âmbito, das infra-estruturas e da criticidade, são implementados diferentes mecanismos de autenticação e por vezes com diferentes níveis, que podem ser a combinação de dois factores, como por exemplo, a palavra-chave do utilizador (algo que a identidade sabe) com algo que tem (o 'token' ou o cartão) ou algo que se é (biometria) como forma de validação da sua identidade.

A autenticação é um processo fundamental num sistema de controlo de acessos, tendo em conta que permite o acesso (sejam eles a dados ou a instalações) para que o utilizador foi autorizado.

2.3. Autorização

Autorização é o processo ou função de permitir o direito de acesso a um recurso. Está relacionado sobretudo com a segurança da informação e dos sistemas, em geral e com o controlo de acessos, em especial.

O processo de autorização implica a definição de políticas de acesso e das regras de controlo de acessos. Com base nessa informação, são usadas regras de controlo de acesso para decidir se os pedidos de acesso devem ser aprovados (concessão do acesso) ou reprovados (rejeição do pedido) e com que tipo de privilégios.

Existem diversas formas de implementar os fluxos de aprovação, podendo ser constituídas listas de controlo de acessos ou controlo de acessos baseados em perfis.

Pode ser igualmente considerado como o processo que permite que um utilizador após autenticação execute acções específicas, e desta forma é visto como o processo posteriori à autenticação.

2.4. Gestão de acessos

Relacionado com a gestão de ciclo de vida de um acesso, refere-se a todo o conjunto de processos e tecnologias que permitem controlar, manter e actualizar as identidades digitais. A gestão de ciclo de vida de identidade inclui o provisionamento (atribuição do acesso) da identidade, a sincronização das identidades na infra-estrutura, a gestão corrente dos atributos, das credenciais e dos direitos da identidade, e a retirada do provisionamento (eliminação do acesso).

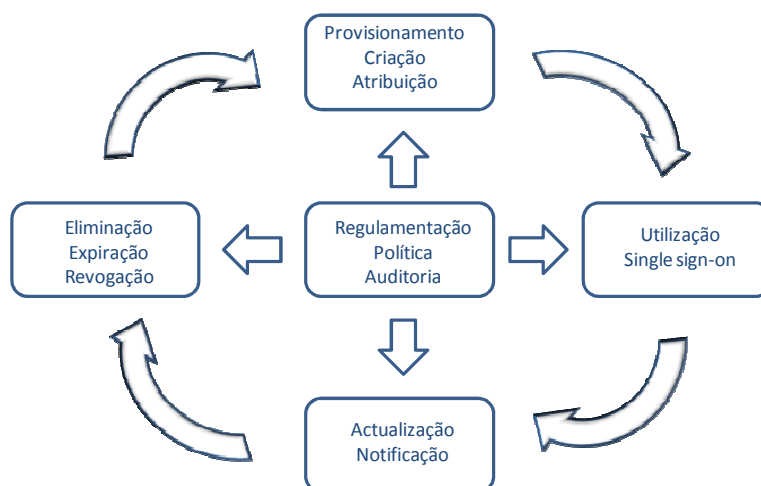


Fig. 8 – Ciclo de vida de uma Identidade

Este processo engloba igualmente implementar segurança, monitorizar os acessos e responder às auditorias ao permitir documentar o ciclo de vida de um acesso, as regras de acesso, o acesso das identidades e o cumprimento com a regulamentação existente.

2.5. Sistema de Gestão de Identidades

A Gestão de Identidades (ou Gestão de ID ou *IdM - Identity Management*) é o termo utilizado para referirem o processo de identificação, autorização e autenticação nos sistemas informáticos. Define-se como um conjunto de processos e tecnologias para o tratamento e manipulação de identidades, que inclui questões sobre a atribuição de identidades a utilizadores, protecção dessa identidade e as tecnologias de suporte à protecção. Um sistema de gestão de identidades tem como objectivo estabelecer e gerir os papéis e os privilégios dos utilizadores assim como controlar o acesso do utilizador à informação crítica dentro da empresa.

Em traços gerais, um sistema de Gestão de Identidades deve ter as seguintes preocupações:

- a) Estabelecer a identidade:
 - a. Relacionar um nome ou número com o sujeito ou objecto;
 - b. Restabelecer a identidade (i.e., relacionar informação adicional ou mesmo novo nome ou número, com o sujeito ou objecto);
- b) Descrever a identidade:
 - a. Possibilidade de atribuir um ou mais atributos aplicáveis a determinado assunto ou objecto;
 - b. Reescrever a identidade (i.e. alterar um ou mais atributos aplicáveis a um determinado assunto ou objecto);
- c) Acompanhar a actividade da identidade:
 - a. Registo da actividade da identidade ou das relações entre identidades.
 - b. Possibilidade de estabelecer padrões de comportamento das identidades.
- d) Eliminar a identidade:
 - a. Possibilidade de eliminar a identidade e a informação / atributos que compõe a mesma.

2.6. Produtos IdM

2.6.1. Microsoft Forefront Identity Manager

A *Microsoft Forefront Identity Manager*ⁱⁱ (FIM) é a solução da Microsoft, para gerir identidades, credenciais e perfis baseados em políticas de acesso.

É apresentado como uma ferramenta abrangente e transversal à organização, com interfaces consoante o tipo de utilizador, de ser uma ferramenta extensível e que suporta ambientes heterogéneos.

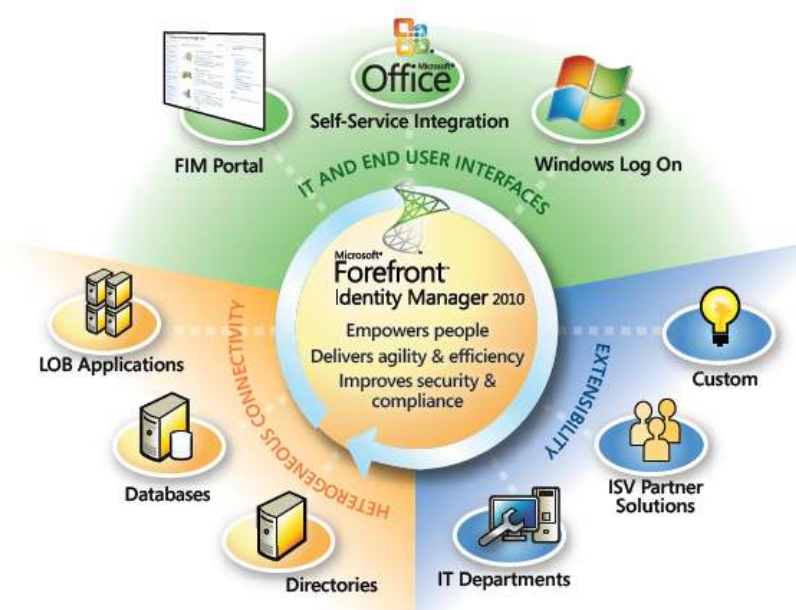


Fig. 9 – Microsoft Forefront Identity Manager (FIM)

Fonte: <http://www.microsoft.com/en-us/server-cloud/forefront/identity-manager.aspx>

São referidos os seguintes benefícios:

Habilita diferentes tipos de utilizadores. Dirigido a utilizadores sejam eles utilizadores finais, profissionais de TI ou programadores. Tem como objectivo principal aumentar a produtividade e diminuir os custos com o *help desk*, entre outros.

É a solução para utilizadores finais ao integrar-se com as aplicações habituais utilizadas por estes utilizadores (*end-users*), nomeadamente com produtos *Office* e o *Windows*®. Permite que o utilizador faça a gestão das suas *passwords* (*reset* das palavras-chave e/ou

do *PIN* dos cartões inteligentes (*smart cards*). Possibilidade de actualizar o seus dados, criar grupos de endereços de e-mail, adicionar outros utilizadores aos grupos sem necessitar de recorrer aos serviços de *help desk*.

É a solução para profissionais de TI ao permitir uma gestão de identidades mais eficiente através do acesso à consola administrativa *SharePoint*®. Disponibilização de um menu via consola que permite criar políticas, fluxos de gestão de contas e acessos transversais à organização.

É a solução para programadores ao apresentar-se como uma ferramenta extensível, ou seja, que permite customizar funcionalidades de acordo as necessidades organizacionais e com a vantagem de integrar-se com produtos actuais de programação como o *Microsoft Visual Studio*® e *.NET*.

Proporciona uma maior agilidade e eficiência, através da automação de actividades associadas à gestão de identidades, o que por sua vez proporciona a redução de custos e dos riscos inerentes. O *FIM* permite a gestão automática de utilizadores, grupos e outros recursos baseados nas políticas da organização e disponibiliza as ferramentas necessárias baseadas em interfaces *Web*, para apoiar os utilizadores finais na construção das suas identidades.

Suporta ambientes heterogéneos e providencia num único ponto a gestão das diferentes identidades, nomeadamente a informação dos sistemas operativos, e-mail, base de dados, directorias, aplicações.

Integra com as ferramentas padrão das infra-estruturas organizacionais, como por exemplo: *Active Directory*® *Domain Services*, *Microsoft Exchange*, e *Active Directory Certificate Services*, o que facilita a implementação do produto.

Aumenta a segurança ao permitir adequar a organização em conformidade com as normas internacionais. Implementação de uma política unificada que permite efectuar a gestão de identidades, de credencias, de recursos e de obter a informação/evidências para auditorias.

Implementação de fluxos de permissões e de delegação, o que permite aumentar o controlo e diminuir o risco associado às actividades de autorização.

Activação de processos de auditoria com o intuito de reduzir os riscos de não-conformidades. Possibilidade de auditar a normas e os eventos processados, e de forçar/implementar as normas que permitem que uma organização esteja em conformidade, de forma automática.

Em resumo, o *Microsoft Forefront Identity Manager* apresenta-se com um produto que actua nas seguintes áreas:

Gestão de Utilizadores – Ferramenta de provisionamento e de de-provisionamento (desactivação) de utilizadores. Constituição de regras para o provisionamento ou de-provisionamento de utilizadores via interface *Web*. Possibilidade de actualização dos perfis pelo próprio utilizador.

Gestão de Credenciais – Gestão de credenciais para os diferentes tipos de utilizadores, nomeadamente constituição de um ciclo de vida integrado com o provisionamento. Na mesma ferramenta é possível gerir o processo de provisionamento e o processo de credenciais.



Fig. 10 – *Microsoft Forefront Identity Manager (FIM)* - áreas de actuação

Fonte: <http://www.microsoft.com/en-us/server-cloud/forefront/identity-manager.aspx>

Gestão de Acessos - Possibilidade de criar grupos com as mesma características e dessa forma evitar uma gestão de identidades repetitiva. Possibilidade de associar o(s) utilizador(es) a grupos com determinados perfis e políticas de segurança. Aumenta a produtividade, melhora a segurança e permite uma implementação em conformidade.

Gestão de Políticas – Estabelecimento e implementação do conjunto de políticas comuns a todos os sistemas da empresa, com a possibilidade de se auditar as mesmas.

2.6.2. Oracle Identity Manager

O *Oracle Identity Manager*ⁱⁱⁱ (*OIM*) é um componente da solução *Oracle Identity and Access Management Suite* desenvolvida pela Oracle.

É apresentado como uma solução de gestão de identidades altamente flexível e escalável conferindo eficiência operacional ao providenciar uma administração centralizada e a completa automação de identidades, de aprovisionamento de utilizadores e dos eventos que ocorrem quer nos sistemas, quer nas aplicações. Possibilidade das organizações automatizar o processo de criação, actualização e eliminação de utilizadores e atribuírem níveis de privilégios aos diferentes recursos da empresa.

Desenvolvido numa arquitectura *JAVA EE* confere-lhe a escalabilidade, uma vez que assenta numa plataforma padrão, robusta e segura que encontra-se na base de muitas das aplicações usadas hoje em dia nas empresas. A possibilidade do *Oracle Identity Manager* relacionar-se com as exigências mais complexas das TI e do negócio sem necessidade de proceder-se a alterações à infra-estrutura existente, às políticas ou a procedimentos implementados, atribui-lhe por sua vez a característica de ser um produto flexível e adaptável à estrutura da organização.

Construído por camadas que consideram a administração das identidades, o aprovisionamento de acordo com as funções e o fluxo de aprovações, tendo em consideração as constantes mudanças organizacionais, que costumam ter um forte impacto na gestão de processos sobre identidades.

São atribuídas as seguintes características ao produto *Oracle Identity Manager*:

Automação da actividade de gestão de identidades que por sua vez permite aumentar os níveis de produtividade, de satisfação dos utilizadores e otimizar as TI.

Gestão de identidades e de delegação da administração dos utilizadores transversal à organização, que se traduz na redução dos custos e no aumento dos níveis de segurança.

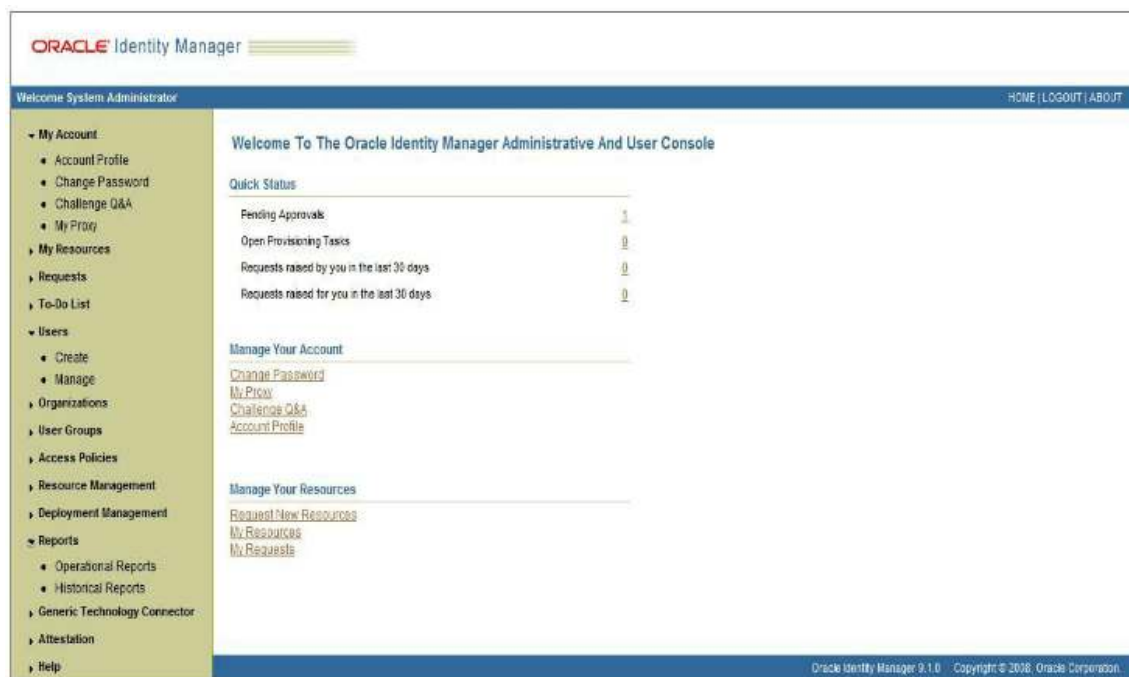


Fig. 11 – Oracle Identity Manager (OIM) – Consola

Fonte: <http://www.slideshare.net/thiago.gutierri/oracle-identity-manager>

Implementação de fluxos de aprovação orientada às políticas de provisionamento, melhorando a eficiência das TI e garantindo a aplicação dos procedimentos em conformidade com as normas.

Gestão de *passwords* efectuada pelos próprios utilizadores, diminui os problemas resultantes de bloqueios, traduzindo-se em melhorias dos níveis de serviço e nos custos com as actividades de *help desk*.

Integração a tecnologia *standard* e com conectores pré-configurados, que permite uma integração rápida e de baixo custo.

Benefícios com a utilização do *Oracle Identity Manager*:

Aumenta a segurança, ao permitir a aplicação de políticas de segurança transversalmente em toda a infra-estrutura, eliminando potenciais falhas na segurança, adjacentes a actividades/procedimentos incorrectos, acessos expirados e/ou não-autorizados.

Processos em conformidade com as normas regulatórias. Implementações de acordo com os requisitos mandatários regulados por exemplo pelo *Sarbanes-Oxley* [SOX, 2006], *21 CFR Part 11* [21 CFR Part 11 (1999-2009)], *Gramm-Leach-Bliley* [Gramm-Leach-Bliley (2001)], *HIPAA* [HIPAA (2011)] e *HSPD-12* [HSPD-12 (2012)], associados à identificação de perfis e identificação das identidades com acesso a informação considerada sensível / confidencial.

Operações simplificadas permitindo a redução de ineficiências e a melhoria dos níveis de serviço e da automação das tarefas rotineiras de administração de utilizadores.

Melhor tempo de resposta do negócio, ao permitir um acesso mais rápido e activo aos sistemas e aplicações.

Redução de custos de TI com a utilização eficiente e transversalmente por toda a infra-estrutura da organização.

Em termos de arquitectura, são-lhe atribuídas as seguintes características:

Simples de implementar, possui um gestor de instalação de auxílio ao processo de migração, integração e configuração com os diversos ambientes.

Flexível, podendo ser instalado numa única instância ou em múltiplas instâncias. A instalação em múltiplas instâncias é mais vantajosa no sentido de permitir otimizar as opções de configuração, maior tolerância a falhas, redundância (disponibilidade), possibilidade de *fail-over* e de balanceamento (possibilidade de mover alternadamente) em situações de carga no sistema.

Aproveitamento da infra-estrutura existente, ao ser projectado numa arquitectura aberta permite a integração com o *software* ou *middleware* existente na infra-estrutura das organizações.

Arquitectura modular, com diferentes níveis de abstracção que toleram alterações sem afectar as definições e estruturas lógicas anteriores, permitindo que as mesmas mantenham-se activas.

Baseado em padrões, incorpora produtos *standard* tais como *J2EE*, e os padrões elaborados pela *Organization for the Advancement of Structured Information Standards (OASIS)* [OASIS, 2012].

2.6.3. Tivoli Identity Manager

O IBM Tivoli®Identity Manager^{iv} é a solução da IBM para a gestão de identidades.

Com o aumento do número de utilizadores e das permissões para o acesso à informação através da utilização de diferentes aplicações, sistemas, ou serviços, a IBM considera que as actuais organizações enfrentam três grandes desafios:

- Implementar requisitos de conformidade internos e regulatórios.
- Manter uma postura segura e eficaz.
- Obter o retorno mensurável do investimento.

Considerando os desafios acima, a IBM desenvolveu o *Tivoli®Identity Manager* que providencia uma solução fácil de implementar e de utilização simples para a gestão de acessos e de identidades e que permite alcançar níveis de segurança e implementar regras de gestão em toda a infra-estrutura de TI.

Através do uso de perfis, contas e permissões de acesso, é possível automatizar a criação, modificação e eliminação dos privilégios dos utilizadores durante o ciclo de vida dos mesmos.

A solução *Tivoli®Identity Manager* disponibiliza:

A criação de uma hierarquia de funções que simplifica a administração, proporciona visibilidade dos acessos do utilizador e ajuda a preencher a lacuna entre os recursos das TI existentes na organização e a implementação dos direitos de acesso dos utilizadores a esses recursos.

Um serviço *Web* para gerir regras de negócio, contas de utilizadores, membros de grupos e palavras-chave.

Gestão de grupos com o intuito de simplificar e reduzir o custo com a administração de utilizadores, oferecendo a capacidade de adicionar, remover ou alterar os atributos de uma entidade do grupo através da consola *Tivoli Identity Manager*.

Fluxos integrados para a submissão e aprovação automática das solicitações dos utilizadores e para as certificações (revalidações) periódicas dos direitos de acesso dos utilizadores.

Processo de provisionamento robusto de atribuição ou remoção dos direitos de acesso do utilizador de acordo com o posicionamento e funções do utilizador na organização e/ou de acordo com regras de negócio.

Um conjunto de controlos que melhoram a segurança, incluindo a separação de funções e a constituição de um ciclo que detecta e corrige as alterações necessárias nos sistemas alvo.

Uma solução de suporte para a gestão dos acessos do utilizador e das palavras-chave nas aplicações e nos sistemas, além de permitir uma solução de integração para a gestão de aplicações personalizadas.

A construção de relatórios flexíveis sobre os direitos de acesso do utilizador, aproveitando a sincronização automática de dados de diferentes repositórios.

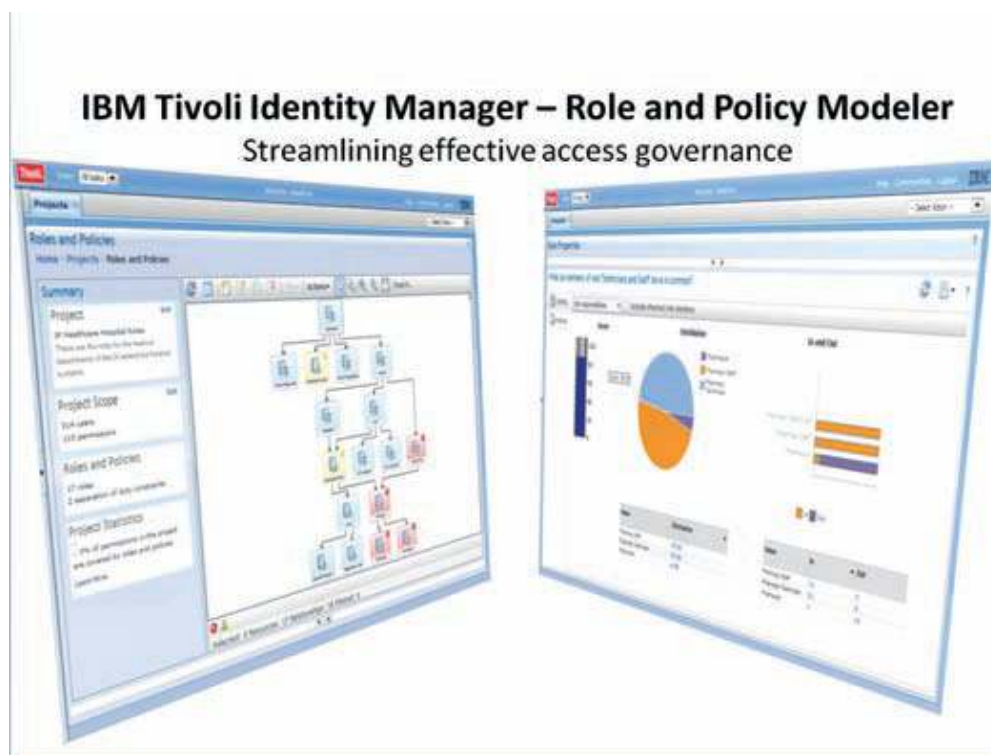


Fig. 12 – Tivoli Identity Manager

Fonte: <ftp://public.dhe.ibm.com/common/ssi/ecm/en/tid10294usen/TID10294USEN.PDF>

As áreas de actuação/gestão efectuadas pelo *Tivoli Identity Manager* são:

- a) Contas de utilizadores, serviços e atributos;
- b) Palavras-chave;
- c) Membros dos grupos;
- d) Acessos;
- e) Gestão de sistemas e aplicações.

São atribuídos ao *Tivoli Identity Manager* os seguintes benefícios:

Permite a redução de custos e implementações em conformidade ao automatizar a criação, modificação, re-certificação e eliminação das contas e dos privilégios dos utilizadores durante o ciclo de vida dos mesmos.

Simplifica a implementação e a validação dos papéis e respectivos acessos em toda a infra-estrutura da empresa.

Gere e previne os conflitos relacionados com a segregação de funções, separando as situações de validação (autorização) das situações de aplicação (execução).

2.7. Sucesso Vs Insucesso dos produtos *IdM*

Produtos muito idênticos, procuram responder às necessidades das organizações em possuírem um *software* de Gestão de Identidades abrangente e universal. Verifica-se, por parte dos fornecedores, a preocupação que o produto permita integrações com outros *softwares* padrões em uso nas organizações. No entanto essa preocupação por si só não chega para garantir o sucesso do produto. Não se trata de simplesmente instalar e relacionar o produto a uma infra-estrutura local, apesar de ser essa a abordagem tradicional, ou seja, o de implementar uma infra-estrutura adicional (baseado num directório de serviços ou num meta directório ou suportados em produtos de *single-sign on*) sobre os ambientes/aplicações existentes.

É fundamental que haja um entendimento da arquitectura e da infra-estrutura da organização para desenvolver e adaptar um produto *IdM* à realidade da empresa. Para isso é imprescindível o envolvimento dos gestores de TI, dos analistas e dos técnicos e a sensibilização das pessoas da organização para a mudança. A introdução de mais um produto no dia a dia das pessoas com determinadas características, se não for abrangente a todas as aplicações constituirá um problema de adaptação.

Por outro lado a grande tendência com a instalação destes produtos é o de não considerar o “legado histórico”, ou seja, as situações actuais ou mais antigas que existem nas organizações e implementar a solução somente para as situações futuras, criando-se variadíssimas formas de actuação que mais cedo ou mais tarde, levam a inconsistências e a erros na infra-estrutura e/ou nos sistemas.

O insucesso destes produtos deve-se sobretudo ao fato de não existir uma análise prévia da infra-estrutura, à falta de sensibilização e conhecimento para a adaptação do produto

à realidade da organização, o que geralmente se traduz na instalação de um produto com falhas e inadequado às mudanças constantes inerentes nas organizações.

Outro factor a considerar neste tipo de ferramenta é a sua complexidade e o facto da mesma exigir desenvolvimento na concepção. Nem sempre são envolvidos as áreas e consideradas todas as situações, o que aparentemente parece ser uma solução “chave na mão” e pronta a instalar, acaba por apresentar-se complexa, de difícil implementação e com custos elevados.

O *IdM* apesar de apresentar-se como uma tecnologia, relaciona-se sobretudo com a mudança na cultura da organização. Um projecto *IdM* instalado com sucesso é o resultado da combinação de quatro factores essenciais:

- Escolha pela **tecnologia adequada**, ou seja, a escolha deve ser baseada num estudo prévio sobre a tecnologia que melhor se adapta ao *core business* da empresa e à necessidade da mesma;
- **Definição correcta dos processos**, nomeadamente dos processos de integração, fluxos de aprovações e *workflows* necessários e customizados;
- **Estabelecimento de uma cultura interna e transversal**, com a divulgação massiva sobre o novo método e procedimentos a adoptar, de forma a evitar desvios ao estabelecido;
- **Gestão de Perfis correcta**, com a definição de perfis internos em conformidade com os requisitos internos e normas e/ou políticas regulatórias.

Em resumo, pode-se considerar uma implementação de sucesso quando se verifica uma integração total da empresa com todas as áreas envolvidas.

O trabalho descrito no capítulo seguinte surge devido a sucessivas tentativas de implementação de produtos *IdM* disponíveis no mercado (exemplo do IAM, OIM) e que, até ao momento não foram concluídas com sucesso.

Apesar de existir actualmente várias aplicações na organização que registam os pedidos de acesso, urge criar um sistema que centralize essa informação inequivocamente e que permita responder atempadamente às necessidades dos utilizadores e das equipas de administração de sistemas.

O insucesso das implementações referidas anteriormente deveu-se, na minha opinião, sobretudo à complexidade dos produtos, à falta de *know-how* quanto a fundamentos de sistemas UNIX e sobretudo à falta de definição de requisitos e de objectivos (o que pretendia ser um sistema de gestão de identidades e com tudo o que isso implica tem sido essencialmente uma ferramenta para registo dos utilizadores e para os fluxos de aprovações, deixando de fora a questão da gestão dos utilizadores nos diferentes sistemas e a questão da gestão das *passwords*). Aparentemente adquiriu-se produtos sem consulta às áreas técnicas e verificou-se que a implementação está condicionada sobretudo aos *workflows* das aprovações, sendo necessário desenvolver internamente o *workaround* com os sistemas UNIX, que permita dar continuidade à “suposta” ferramenta IdM.

Considerando esta situação e uma vez que se mantém a necessidade de desenvolver os conectores com os sistemas, foi proposto desenvolver um sistema interno que permita que os utilizadores registem e interagem com os sistemas finais, respeitando o registo de todo o processo num sistema próprio.

3. Gestão de Identidades em Ambientes UNIX

3.1. Gestão de Utilizadores nos sistemas UNIX

Nos sistemas UNIX e à semelhança de um sistema de Gestão de Identidade é possível implementar requisitos essenciais de gestão de utilizadores, nomeadamente:

- a) Ciclo de vida de uma conta – relacionado com a criação, eliminação, activação, desactivação e modificação da conta do utilizador. É possível implementar mecanismos automáticos nos sistemas UNIX que, baseados em determinados parâmetros pré-definidos permitem gerir o tempo de vida das contas dos utilizadores. Esses parâmetros podem ser aplicados nos sistemas individualmente e estão relacionados por exemplo com: as acções após criação das contas de utilizadores, nomeadamente a obrigatoriedade que cada utilizador terá em alterar a *password* atribuída por defeito; o bloqueio da conta do utilizador após um determinado período de inactividade; contas de utilizadores com acessos temporários, isto são contas com um tempo de vida (início e fim) pré-definido.
- b) Gestão de palavras-chave – relacionado com a gestão de *passwords* e com a sua complexidade. Podem ser definidos parâmetros de validação da complexidade das *passwords*, nomeadamente a obrigatoriedade de colocar caracteres numéricos e alfanuméricos; critérios para a selecção das *passwords* reutilizáveis (isto é, definição de histórico de *passwords* permitido); período máximo de utilização das *passwords* (tempo de vida de uma *password*); tipo de encriptação, etc.
- c) Tipo de Acesso – os acessos e os privilégios ao sistema e, ou a aplicações podem ser limitados com a implementação de perfis/grupos.
- d) Gestão de Comandos – possibilidade de restringir/permitir determinados comandos a contas de utilizadores de acordo com o perfil ou privilégios. Em UNIX é configurável através do comando *sudo*.
- e) Registo / Auditoria – relacionado com a activação de ficheiros de *logs*, onde são registados actividades relacionadas com o acesso das contas dos utilizadores, como por exemplo, o registo de entrada (*login*), registo de saída (*logout*), último acesso

(last (un)successful access); data de alteração da *password (last (un)successful password change)*, acessos via *sudo*, etc.

Com a definição destes requisitos por sistema mas com regras transversais é possível criar uma estrutura que apesar de não ficar centralizada no seu todo, pode ser eficaz e traduzir na melhoria da gestão de utilizadores e das respectivas identidades.

Nos capítulos seguintes irá ser documentado a proposta de um SI Gestão de Identidades baseado na linguagem *UML*, enquanto linguagem de especificação, construção, visualização e documentação considerando os aspectos conceptuais e/ou concretos do sistema.

A linguagem *UML* foi aprovada em 1997, como *standard* pelo *Object Management Group (OMG [1997-2012])*, um consórcio internacional de empresas que define e autentica padrões na área de Orientação a Objectos e tem sido adoptado por diversas empresas e instituições para a modelação de software e de negócio.

3.2. Solução SI Gestão de Identidades

3.2.1. Linguagem UML

A versão UML 2.0 está organizada em treze tipos de diagramas segundo três categorias:

- **Diagramas estruturais** (*Structure Diagrams*), que incluem Diagrama de classes (*Class Diagram*), Diagrama de objectos (*Object Diagram*), Diagrama de componentes (*Component Diagram*), Diagrama de estrutura composta (*Composite Structure Diagram*), Diagrama de pacotes (*Package Diagram*) e Diagrama de instalação (*Deployment Diagram*).
- **Diagramas comportamentais** incluem Diagrama de casos de utilização (*Use Case Diagram*), Diagrama de actividades (*Activity Diagram*) e Diagrama de máquina de estados (*State Machine Diagram*).

- **Diagramas de interacção** (*Interaction Diagrams*), que derivam dos diagramas comportamentais, incluem Diagrama de sequência (*Sequence Diagram*), Diagrama de comunicação (*Communication Diagram*), Diagrama temporal (*Timing Diagram*) e Diagrama de visão geral da interacção (*Interaction Overview Diagram*).

De acordo com Alberto Silva e Carlos Videira [Silva, A e C.Videira, (2005), p.108], existem diagramas principais que permitem obter uma visão completa do sistema que se pretende implementar, podendo ser construídos de forma independente e ou paralela, mas mantendo o nível de integração, consistência e coesão entre eles. O ponto de partida de cada modelo deverá ser sempre o sistema em si. São eles:

- Diagramas de casos de utilização, que representam a visão do sistema tendo em conta a perspectiva dos utilizadores;
- Diagramas de classes e diagramas de objectos que permitem especificar a estrutura estática de um sistema segundo uma abordagem orientada por objectos;
- Diagramas de sequência e diagramas de comunicação, enquanto representativos de diagramas de interacção entre os objectos, diagramas de estados e diagramas de actividades, como forma de especificar a dinâmica ou comportamento do sistema segundo uma abordagem orientada por objectos;
- Diagramas de componentes e diagramas de instalação, como forma de obter-se a disposição dos componentes físicos (*software e hardware*) de um sistema.

Para a construção dos diagramas apresentados de seguida foi utilizado o *software UML* da Astah Community [Astah (2011)], versão 6.5.1 (*Model Version: 35*).

3.2.2. Diagrama de casos de utilização (*Use Case Diagram*)

Um diagrama de casos de utilização permite representar graficamente o resultado da análise de requisitos de um sistema e descrever a relação entre os actores (sejam eles pessoas ou objectos) e os casos de utilização de um dado sistema. Com este diagrama pretende-se obter uma visão global e de alto nível do sistema, sendo essencial a correcta definição da sua fronteira.

Sendo assim, o objectivo do sistema de Gestão de Identidades é que os utilizadores de uma determinada empresa através de um portal interno possam efectuar pedidos de provisionamento aos sistemas, desbloquear e activar os seus acessos, validar em que sistemas já possuem acesso, solicitar alteração de perfil e consultar o estado do pedido.

Este sistema tem como pressuposto a atribuição de uma conta de acesso ao sistema de permissões de acesso (baseado por exemplo na *Active Directory*¹), estando o acesso ao portal limitado a essa autenticação.

Pretende igualmente possibilitar actividades de revalidação de acessos, por um funcionário com perfil específico (auditor interno) com o intuito de proceder à eliminação das contas de utilizadores desactivadas.

¹ *Active Directory* (AD) é um serviço de directórios desenvolvido pela Microsoft para redes de domínio Windows. Incluído na maioria dos Sistemas Operativos Windows, o *Active Directory* funciona como um ponto centralizador para a administração de rede e de segurança. Pode ser usado para autenticar e autorizar os utilizadores e computadores dentro de uma rede de domínio Windows, para atribuir e aplicar políticas de segurança e/ou instalar ou atualizar *software* nos computadores da rede. O *Active Directory* utiliza *Lightweight Directory Access Protocol* (LDAP) versões 2 e 3, *Kerberos* e *DNS*.

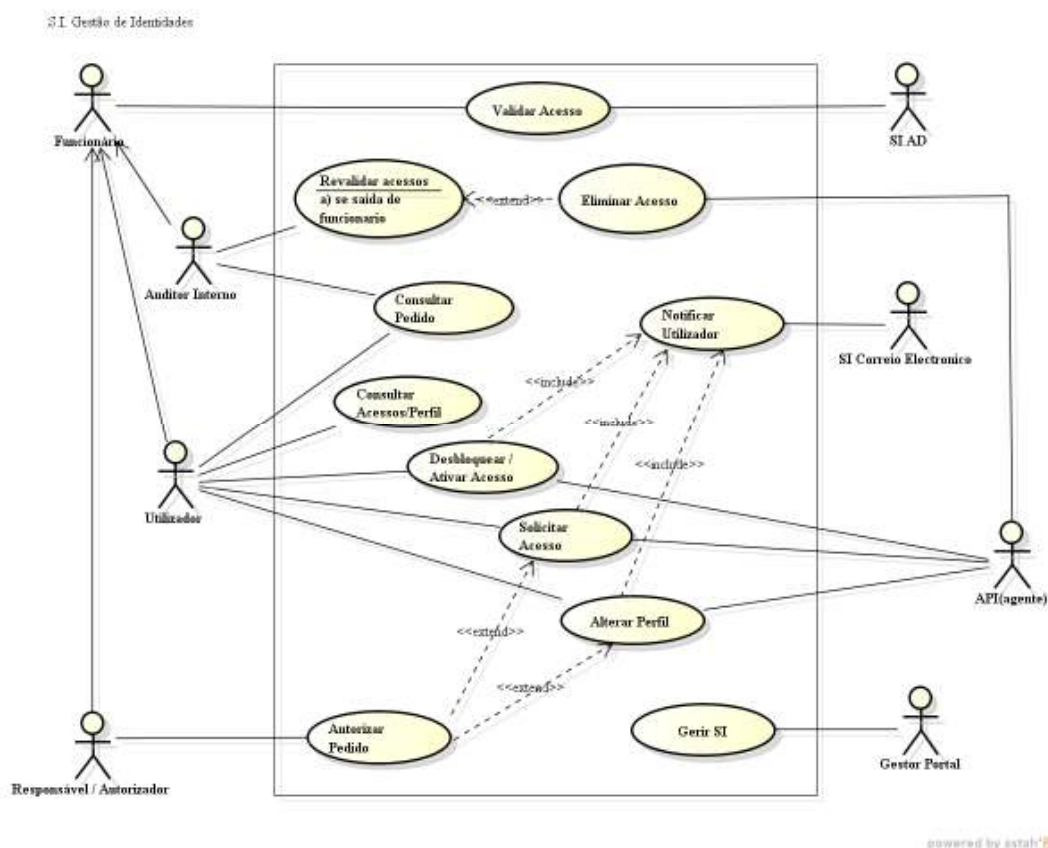


Fig. 13 – Diagrama Caso de Utilização: S.I. Gestão de Identidades

Na figura 13 é representado o S.I. Gestão de Identidades pretendido onde são identificados as seguintes situações:

1. Actores:

Como todos os utilizadores têm de ser autenticados, foi identificada uma generalização entre actores. O actor genérico funcionário representa que, tanto o utilizador como o auditor interno necessitam de validar o seu acesso enquanto utilizadores do sistema através do sistema de permissões de acesso externo ao SI e que pressupõe-se que esteja instalado na empresa (representado pelo actor SI AD). Foi identificado igualmente a necessidade de existir um Responsável/Autorizador, também

funcionário, para aprovações e um Gestor do Portal. Sendo assim temos como actores:

- O utilizador do sistema que é simultaneamente um funcionário e portanto com acesso atribuído pelo sistema de permissões de acesso instalado na empresa.
- O auditor interno que apesar de ser um utilizador do sistema distingue-se pelo tipo de perfil que será atribuído.
- Os SI AD, SI Correio Electrónico e API (agente) são sistemas ou processos externos ao sistema, mas necessários para a interacção e validação dos acessos.
- O Responsável/Autorizador para o *workflow* de pedidos/acções específicas.
- E um Gestor Portal que terá como função administrar o sistema.

2. Use Cases:

Foram identificados os casos de utilização, referentes às actividades base de um utilizador e no seu relacionamento com os sistemas alvo, nomeadamente a possibilidade: Consultar Acessos/Perfil, Desbloquear Acesso, Alterar Perfil e Solicitar Acesso. Foi igualmente identificada a necessidade do utilizador pretender Consultar Pedidos feitos no portal. Foi considerado o caso de utilização Revalidar Acessos, conforme pretendido para a actividade de revalidação dos acessos associada especificamente ao Auditor Interno, e os casos de Notificar Utilizador, Autorizar Pedido, Validar Acesso e Eliminar Acesso, como casos de utilização adicionais ao sistema mas de apoio aos casos de utilização principais.

3. Relações:

Em termos de relações entre os casos de utilização foram estabelecidas as relações onde Solicitar Acesso, Alterar Perfil e Desbloquear Acesso incluem (relações representadas pela seta a tracejado *include*) Notificar Utilizador, ou seja, o Utilizador deve ser notificado por correio electrónico do resultado da acção/fecho do processo; e as relações de dependência (relações representadas pela seta a tracejado *exclude*), que implicam uma interacção implícita de um outro actor. Neste sistema temos os casos Solicitar Acesso e Alterar Perfil com o caso Autorizar Pedido a ser efectuado pelo Responsável/Autorizador, e o caso Revalidar Acessos, na situação de Se verificar saída de funcionário, com o caso Eliminar Acesso.

Por último os casos Solicitar Acesso, Desbloquear Acesso, Alterar Perfil e Eliminar Acesso interagem com o processo externo ao SI, descrito como o actor API (agente) .

3.2.3. Diagrama de classes (*Class Diagram*)

Um diagrama de classes pretende descrever a estrutura de um sistema e em particular as entidades existentes, a sua estrutura interna e o relacionamento entre si. É fundamental descrever correctamente as classes e o tipo de relações.

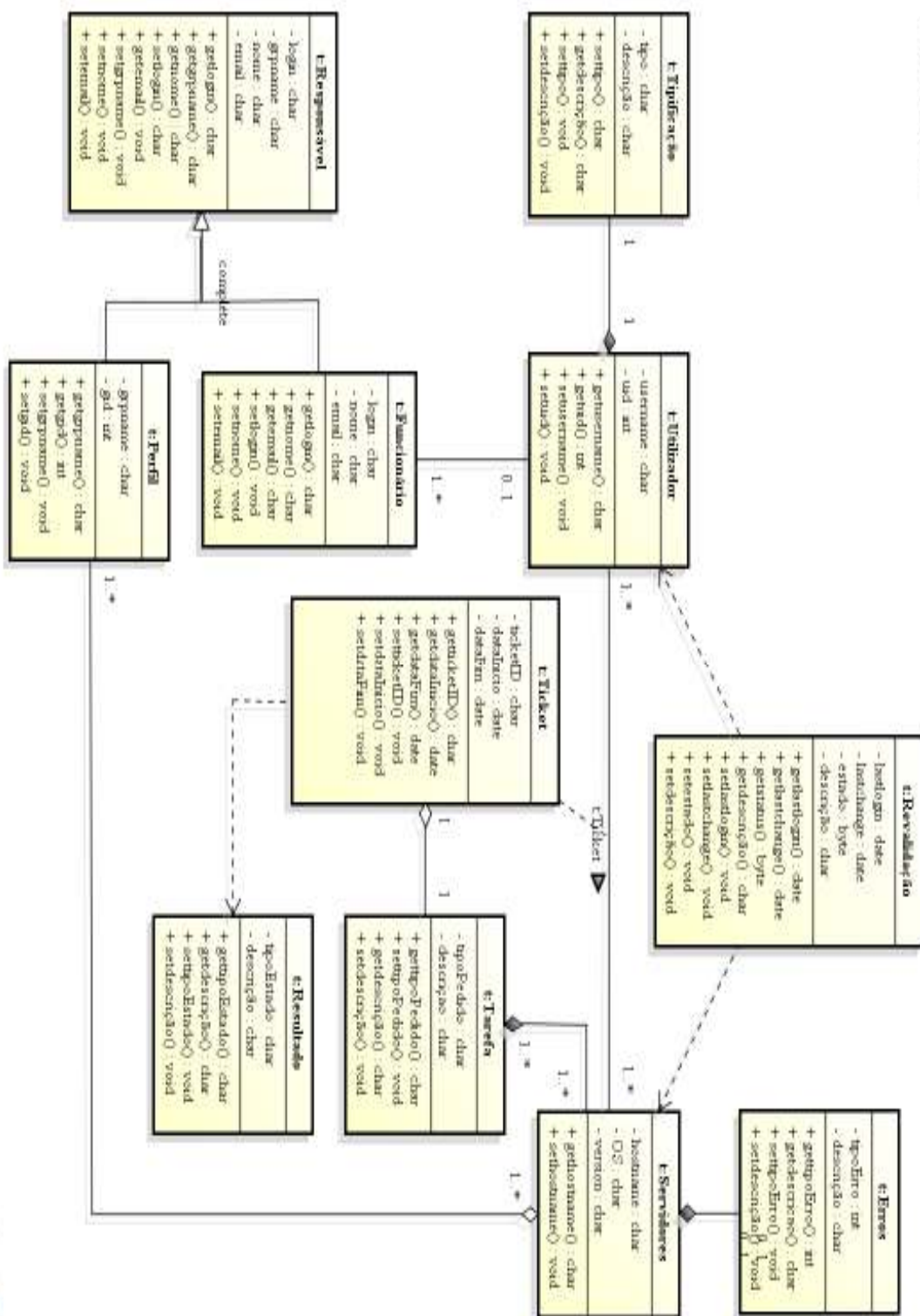


Fig. 14 – Diagrama de classes: S.I. Gestão de Identidades

1. Classes:

As classes, conforme indicado na figura 14, são representadas por rectângulos sendo a 1ª secção o nome da classe, seguido da lista de atributos e por fim da lista de operações.

Considerando o SI Gestão de Identidades e o seu objectivo foram estabelecidas as classes principais: `t:Utilizadores`, `t:Servidores` e `t:Perfil`, com o intuito de construir a base de dados com a informação dos utilizadores, dos servidores alvo e dos perfis/grupos existentes.

Foi igualmente estabelecida a classe genérica (simbologia representado pela expressão *complete*), mais concretamente a classe `t:Responsável`, onde todos os atributos pertencem a uma das classes particulares, ou seja, quer a identificação do responsável, quer a identificação do perfil são obtidas através das classes `t:Funcionário` e `t:Perfil`.

Ao estabelecer o relacionamento das classes principais e do sistema foram criadas outras classes, que de seguida refiro na componente de relações.

2. Relações:

As classes podem-se relacionar através de:

Associações – em que os objectos têm consciência uns dos outros e colaboram entre si. É o caso da classe privada `t:Funcionário` que pretende estabelecer a associação de um funcionário (atributo *login*) com “n” utilizadores (`t:Utilizador`, atributo *username*).

Agregações – onde existe a ilusão de funcionar como um objecto único, e em que o controlo é medido por um único objecto (“o todo”), representado por um losango colocado junto à classe que representa o elemento agregador, ou seja, “o todo”. É o exemplo das classes `t:Perfil` com a classe `t:Servidor`, em que o conceito perfil existe enquanto enquadrado na associação com um servidor e a classe `t:Tarefa`

com a classe `t:Ticket`, onde um tipo de pedido (exemplo: criar utilizador) só é iniciado com a atribuição de um pedido (*TicketID*).

Composições ou agregações fortes – em que “as partes” não podem existir sem “o todo”, e em que “o todo” é responsável pela criação e destruição das suas “partes”. Encontra-se representado por um losango a cheio colocado junto à classe que representa o elemento agregador, “o todo”. Para este tipo de relações foram identificadas a relação das classes `t:Tipificação` com a classe `t:Utilizador` e da classe `t:Erros` com a classe `t:Servidor`, em que a existência destas classes só acontecem enquanto existir a relação tipo de utilizador com utilizador, e tipo de erro versus resultado da acção nos servidores.

Existem igualmente situações de classes associativas, em que ao colocar um atributo numa associação, transforma-se a associação numa classe associativa (representado pela linha a tracejado), ou seja, para relacionar a associação `t:Utilizador` com `t:Servidor` foi considerado a abertura de um pedido, com os atributos de `ticketID`, `dataInicio` e `dataFim` e nesse sentido criou-se a classe associativa `t:Ticket`.

E relações de dependência, em que as classes existem devido a uma relação de precedência entre elementos (exemplo da classe `t:Resultado` do pedido que tem como precedência a classe `t:Ticket`); ou em que a alteração da especificação do elemento-fornecedor implica alteração no elemento-cliente (exemplo da classe `t:Revalidação`, o elemento-cliente com as classes `t:Utilizador` e `t:Servidor`, que representam os elementos-fornecedores. As relações de dependência são representadas por linhas dirigidas a tracejado.

3.2.4. Diagrama de objectos (*Object Diagram*)

Um diagrama de objectos procura descrever o conjunto de instâncias compatíveis com determinado diagrama de classes. Permite pormenorizar os detalhes do sistema em determinado momento, ao apresentar os possíveis cenários de concretização.

Considerando o diagrama de classes descrito acima foi elaborado como exemplo o diagrama de objectos simulando a situação do caso de utilização Solicitar Acesso, considerando os elementos/atributos das diferentes classes.

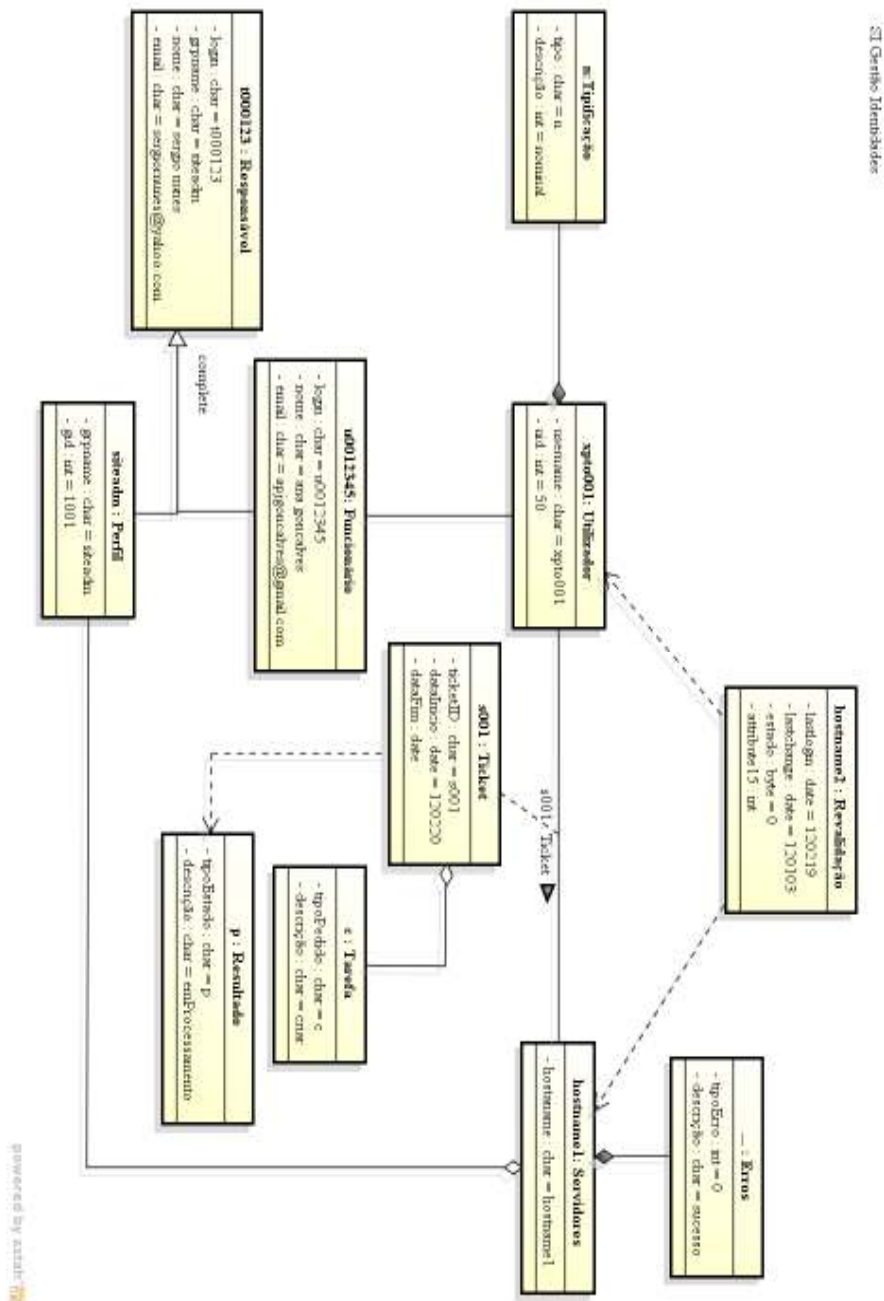


Fig. 15 – Diagrama de Objectos: Solicitar Acesso

3.2.5. Diagrama de sequência (*Sequence Diagram*)

Um diagrama de sequência procura representar as interações entre objectos segundo uma visão temporal. Os objectos são representados por “linhas de vida” e interagem através da sequência de chamada de métodos (troca de mensagens) ao longo de um determinado período de tempo.

Nas páginas seguintes são apresentados os diagramas de sequência para os casos de utilização do sistema. Na figura 16 estão representadas as interações entre o actor *Funcionário*, o actor *SI AD* e o próprio sistema como um todo. O sentido das setas indicam o sentido da interacção entre actores. As setas a cheio em linha contínua representam a invocação de uma operação síncrona e a seta a tracejado corresponde ao retorno da mensagem. Considerando o acesso ao sistema, a resposta aos *inputs* (preenchimento do *login* e da palavra-chave) requerem simultaneidade, ou seja uma resposta imediata, daí ser constituído como processo síncrono.

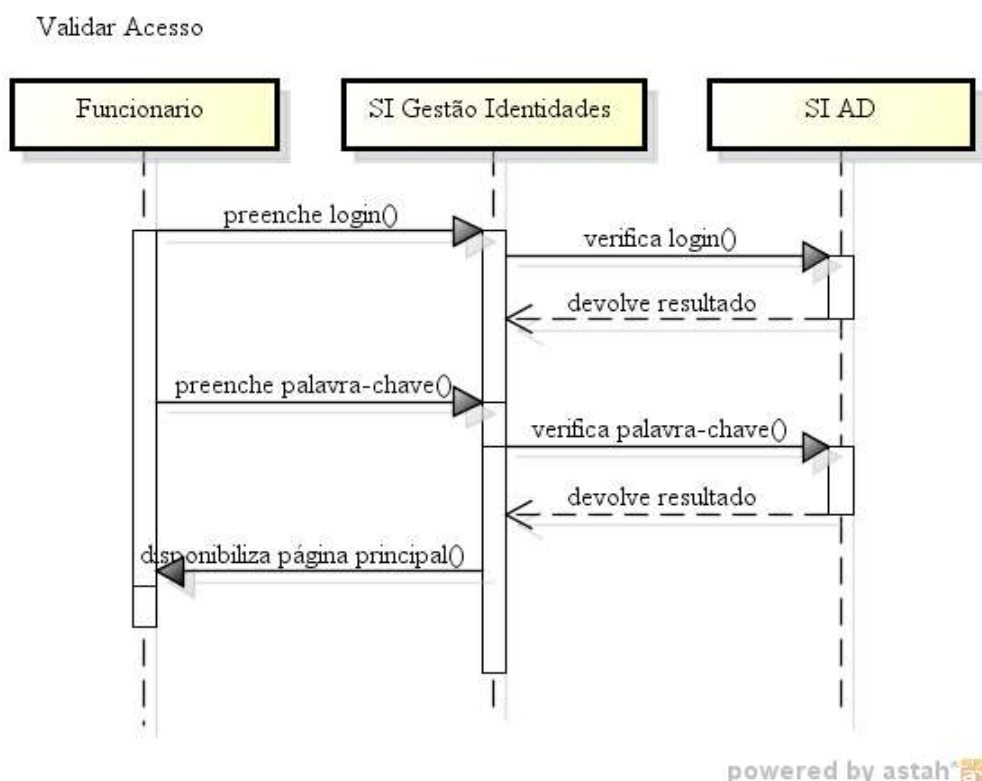


Fig. 16 – Diagrama de sequência: Validar Acesso

Na figura 17, foi desenhado o diagrama de sequência referente aos casos de utilização Solicitar Acesso e Alterar Perfil (inclui Autorizar Pedido) tendo em conta a similaridade dos dois processos.

Foram envolvidos os actores Utilizador, Responsável/Autorizador e API (agente) . Até determinada operação (retorno de que o pedido foi submetido) verifica-se a simultaneidade dos processos, sendo a continuação das operações, após essa fase efectuada em modo assíncrono e dependente dos *inputs* anteriores (autorização concedida e resultado da *API*).

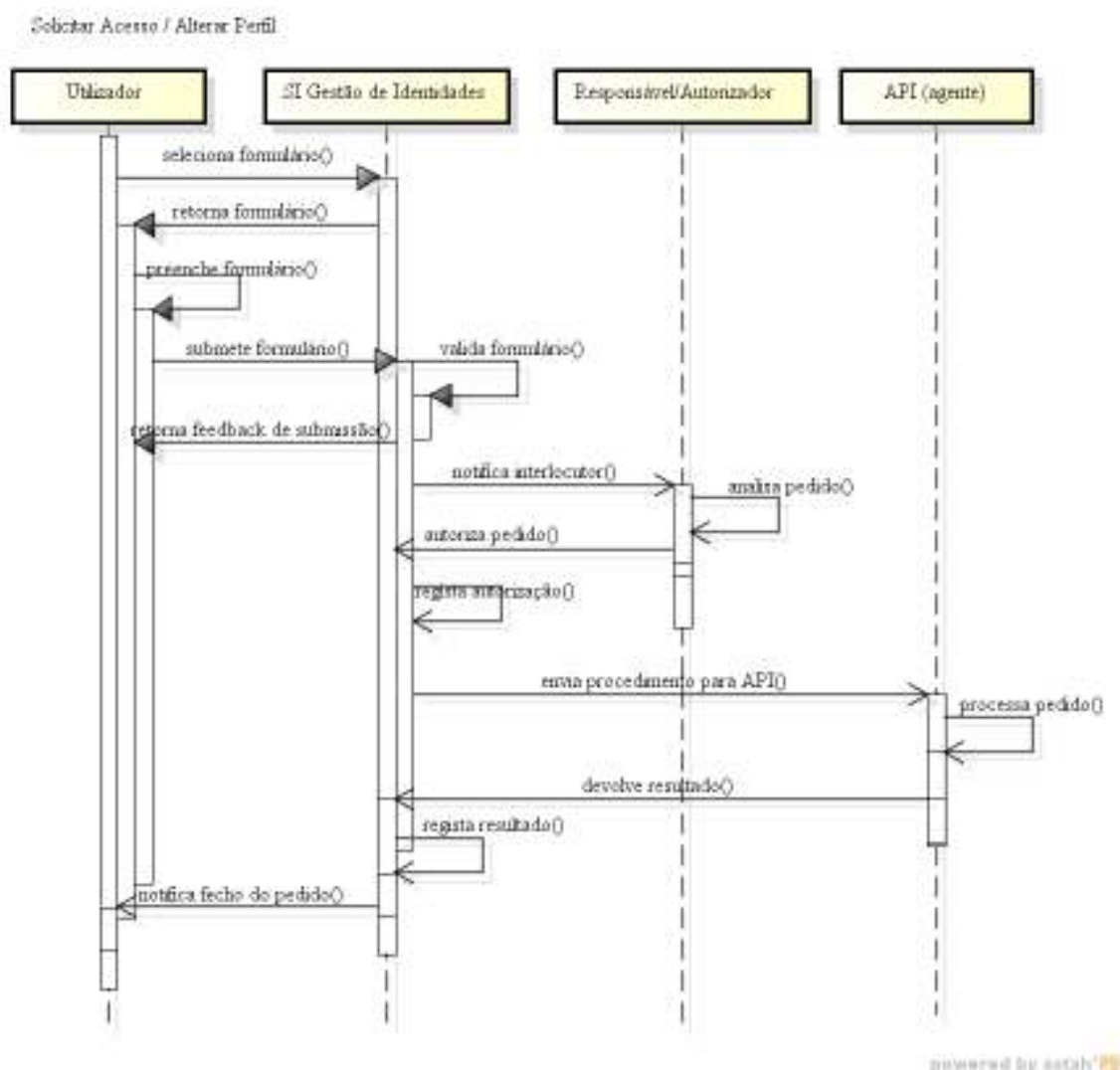


Fig. 17 – Diagrama de sequência: Solicitar Acesso / Alterar Perfil

A sequência dos processos Desbloquear/Activar Acesso e Eliminar Acesso diferem do anterior ao não ser necessário a intervenção do actor Responsável/Autorizador.

Mantêm-se as operações síncronas até à situação de retorno de submissão (que terá como resultado, a atribuição de um número de registo / ticket para o Utilizador) e o desencadear de operações assíncronas com o envio do pedido para o actor API (agente).

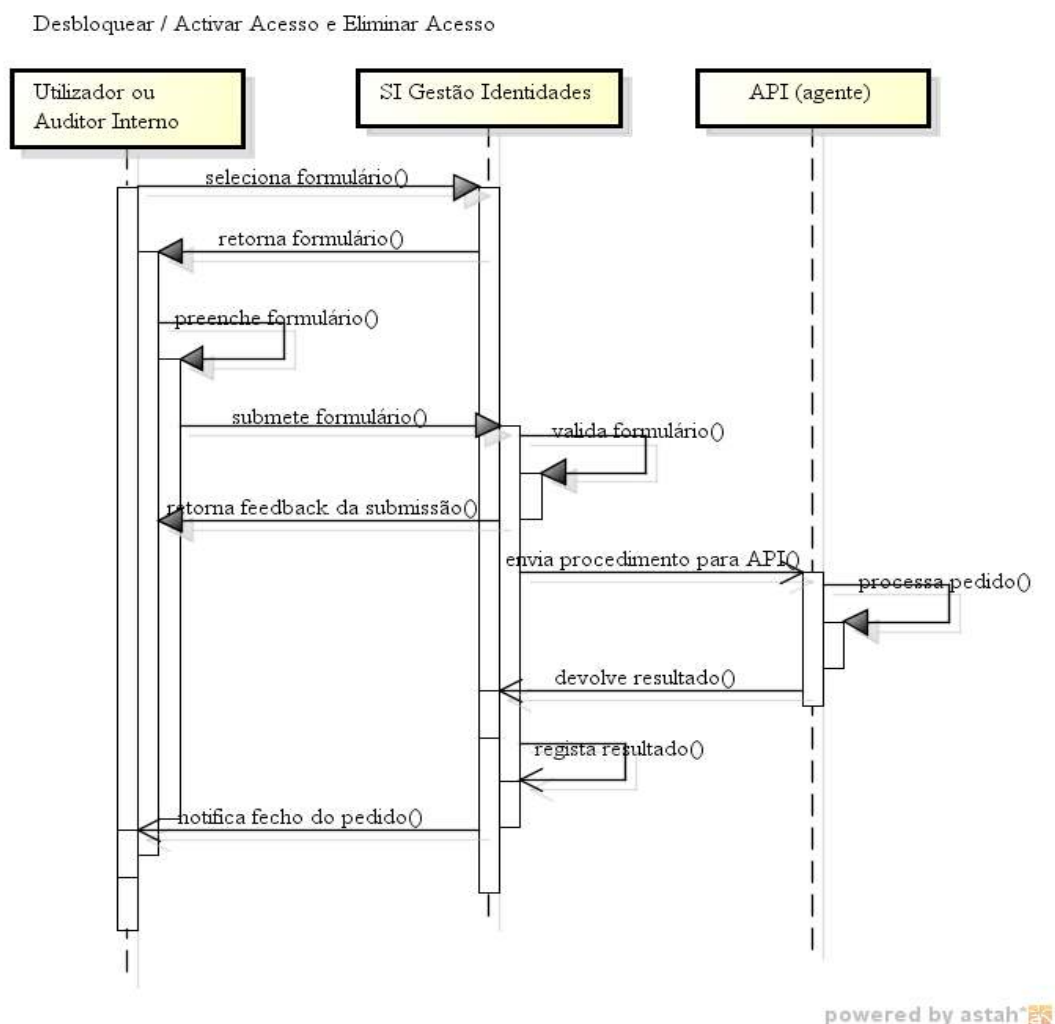


Fig. 18 – Diagrama de sequência: Desbloquear / Activar Acesso e Eliminar Acesso

Para os casos de utilização Consultar Acessos/Perfil, Consultar Pedidos e Revalidar Acesso (figura 19) a interacção entre Utilizador e o sistema é efectuada de modo síncrono, tendo em conta que trata-se basicamente de consultar a informação registada em base de dados.

Estas operações interagem somente entre o actor Utilizador ou Auditor Interno (consoante o caso de utilização) e o sistema.

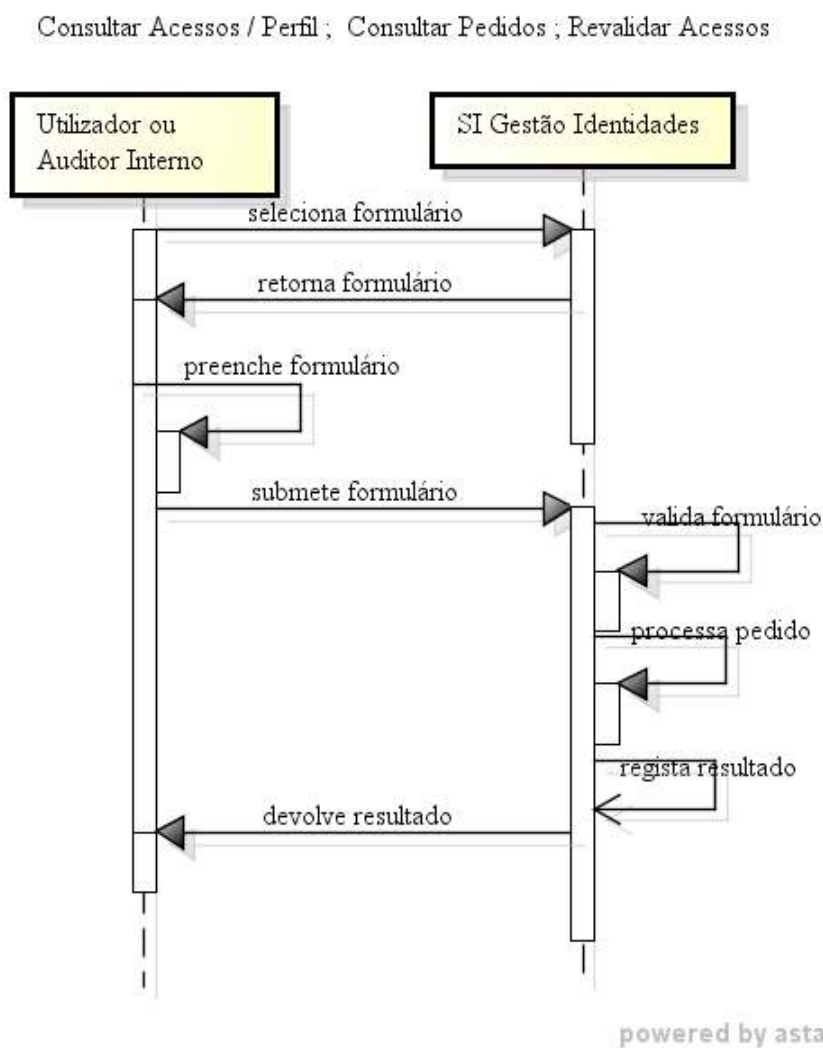


Fig. 19 – Diagrama de sequência: Consultar Acessos / Perfil, Consultar Pedidos e Revalidar Acessos

3.2.6. Diagrama de estados (*State Machine Diagram*)

Um diagrama de estados permite descrever o ciclo de vida de um objecto, subsistema ou sistema global. Descreve as sequências de estado que um objecto ou uma interacção pode passar ao longo do tempo da sua existência em resposta a estímulos recebidos, conjuntamente com as suas respostas e acções.

Na figura 20 é representado o diagrama de estados para a situação *Validar Acesso* e para o estado *Apresentar Ecrã de Acesso*. Quando o sistema é “chamado” para este estado foram identificadas as seguintes situações:

1. Acções de entrada (*entry*): apresentar o Menu de acesso, ou seja, ecrã a solicitar o *login* e a palavra-chave.
2. Actividades (*do*): verificar o preenchimento dos campos.
3. Acções de saída (*exit*): registo dos dados introduzidos pelo utilizador.

Consoante a situação, campos preenchidos ou por preencher, o sistema passa para o estado seguinte (campos preenchidos) ou retoma o ecrã inicial novamente (não foram preenchidos todos os campos). Na situação de campos preenchidos, o sistema assume outro estado que tem como entrada a informação dos campos (*login* e *palavra-chave*), como actividade: ler a informação, como evento: consultar a *AD* e como acção de saída, permitir ou negar o acesso. Consoante o resultado retoma o ecrã inicial (por dados incorrectos / *login* inválido) ou inicia novo estado.

No diagrama foi considerado que o estado *Validar Acesso* é dado como terminado quando se verifica a situação *login* válido.

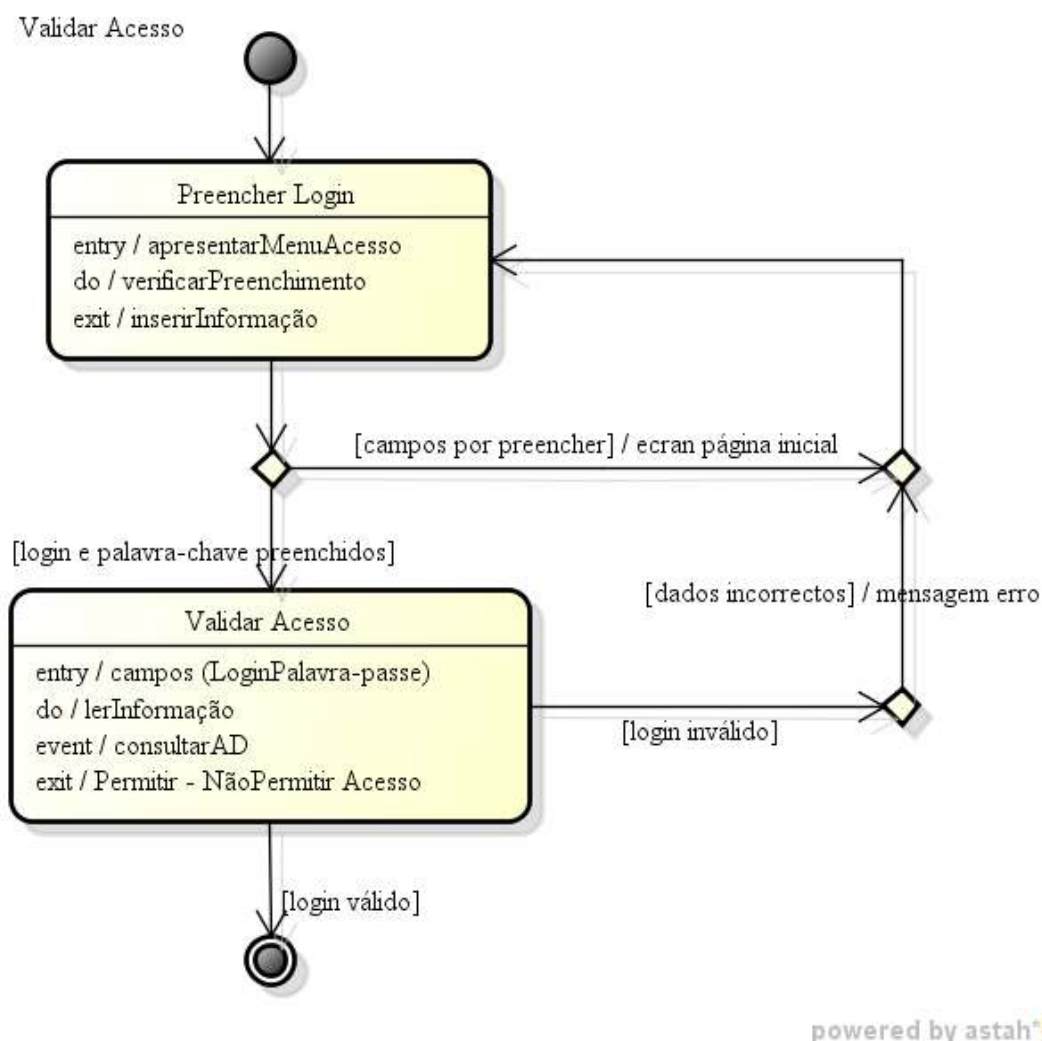


Fig. 20 – Diagrama de estados: Validar Acesso

O diagrama seguinte apresenta o estado *Seleccionar Formulário*, presente em todos os diagramas de sequência. Este é também o estado em que o sistema fica após a autenticação ser efectuada com sucesso. Nesse sentido é identificado o histórico do estado anterior, representado pelo círculo com a letra H.

Foram identificados vários estados com as respectivas situações de incorrecção culminando na situação de que os formulários terminam em três situações possíveis: *Para Executar*, *Para Autorizar* e *Para Consultar*.

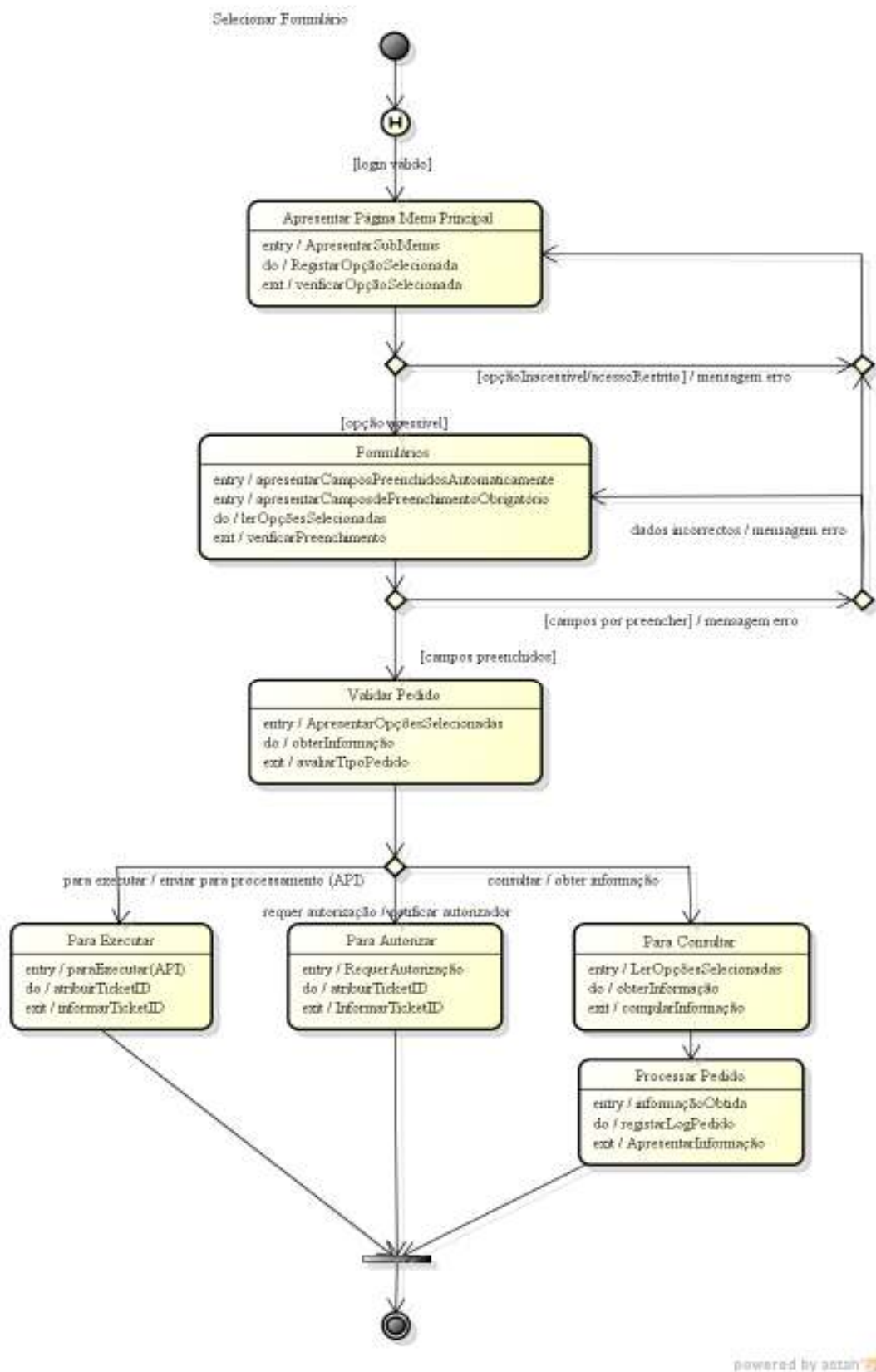


Fig. 21 – Diagrama de estados: Selecionar Formulário

3.2.7. Diagrama de actividades (Activity Diagram)

Um diagrama de actividade permite descrever a lógica dos processos de um sistema ou das suas funções através do encadeamento de operações (actividades e, ou acções) e das decisões que permitem determinar quando e como são realizadas.

A figura 22 representa o fluxo da actividade Validar Acesso desde o início até à sua conclusão, apresentando as entidades / actores intervenientes (Funcionário, o sistema – SI Gestão de Identidades e o SI AD).

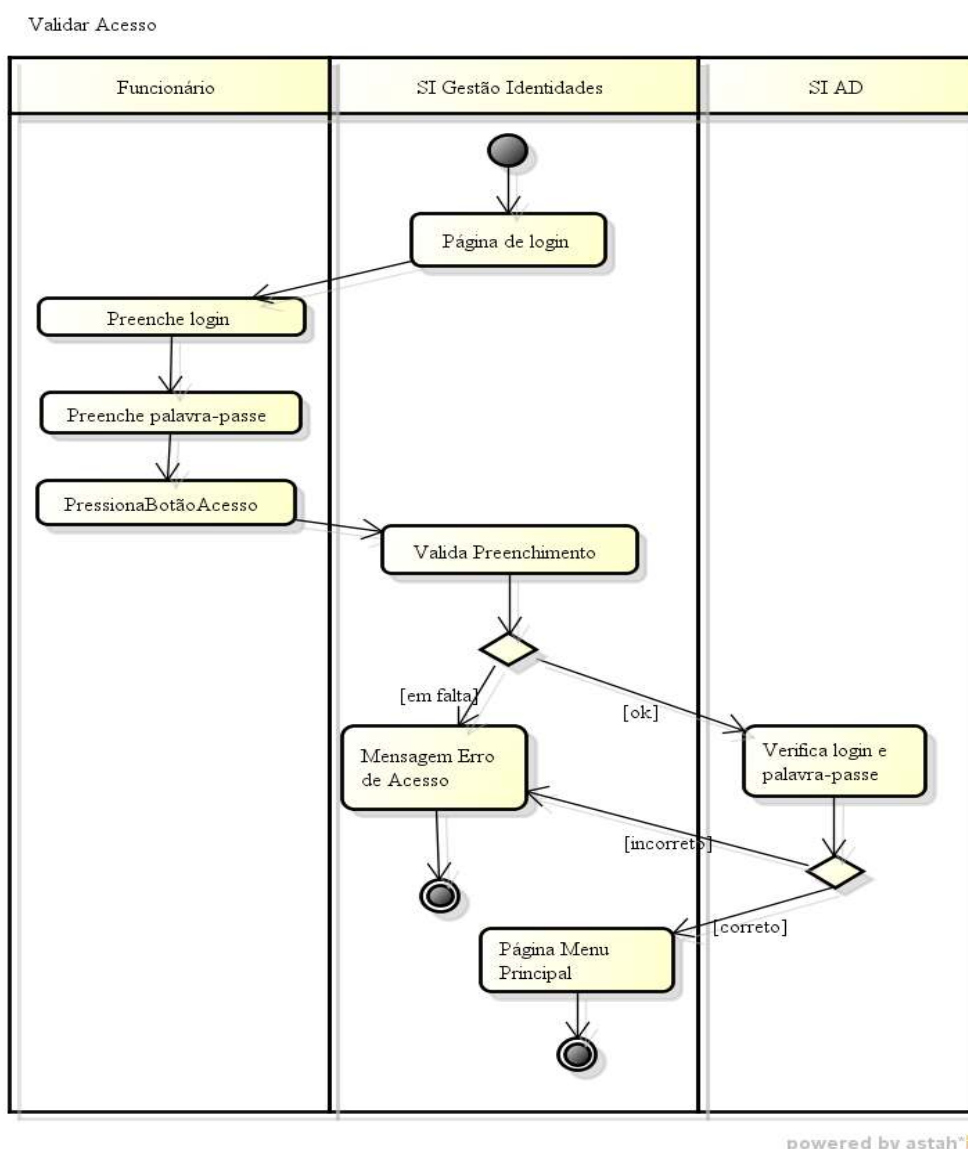


Fig. 22 – Diagrama de actividades: Validar Acesso

O início do fluxo é representado pelo círculo preto, sendo a primeira acção feita pelo sistema ao apresentar a página de *login*. De seguida são representadas o conjunto de operações que cada actor realiza. Dependendo das respostas poderão existir resultados diferentes (representado pelo losango) que por sua vez originam novos fluxos.

No fluxo da actividade *Solicitar Acessos* (figura 23) existem situações de paralelismo das operações realizadas, nomeadamente por exemplo, na operação de após *Registo do Pedido* em que o sistema encaminha o ticket (1) Para *Aprovação*; (2) *Emite e-mail para o Responsável/Autorizador* e (3) *Actualiza o estado do ticket (para Pendente de Autorização)*.

Verificam-se igualmente situações em que a resposta conduz à mesma operação ou seja, na questão da resposta à *Decisão pelo Responsável/Autorizador*, independentemente da mesma (*Rejeita Pedido, Autoriza Pedido*), a operação seguinte será *actualizar o estado do Pedido*.

A diferença entre este tipo de diagramas e o diagrama de sequência é que podemos documentar o fluxo das actividades e, ou operações conforme as respostas dadas. No diagrama de sequência só são representadas a sequências das operações numa visão temporal, o diagrama de actividades permite-nos observar o fluxo das actividades, as operações realizadas e os novos fluxos baseados na tomada de decisões / resultados anteriores.

Na actividade `Solicitar Acesso/Alterar Perfil` o fluxo de actividades passa por diversas operações externas ao sistema. Nestas situações podemos ter dois tipos de comportamentos: situações em que não é necessário o retorno da operação e situações em que é necessário o retorno para continuação das operações. O envio de e-mail para determinado actor não necessita de retorno da operação (é o caso da operação `enviar e-mail para Autorizador` e `e-mail para o Utilizador`). Por sua vez, a operação `encaminha para aprovação` implica o retorno da actividade e nesse sentido há que acautelar o fluxo de retorno das respostas e a operação/actividade seguinte.

Para a situação seguinte e considerando a similaridade das actividades e das operações foi possível representar as actividades `Desbloquear/Activar Acessos` e `Eliminar Acessos` no mesmo diagrama (figura 24) apesar dos actores serem diferentes (`Utilizador` para a actividade `Desbloquear/Activar Acessos`, e `Auditor Interno` para a actividade `Eliminar Acessos`).

Foram identificadas situações de fluxos diferentes consoante o resultado da operação (`verificação incorrecta` ou `obtenção de informação sem resultados`), mas que levam ao mesmo resultado final (ou seja informa erro e concluí a actividade). Na situação `Recebe Ticket é fechado` a actividade correspondente e, é transmitido o resultado ao `Utilizador` ou `Auditor Interno` conforme o caso, no entanto o processo continua no sistema para a actividade seguinte (`Inicia procedimento API (agente)`).

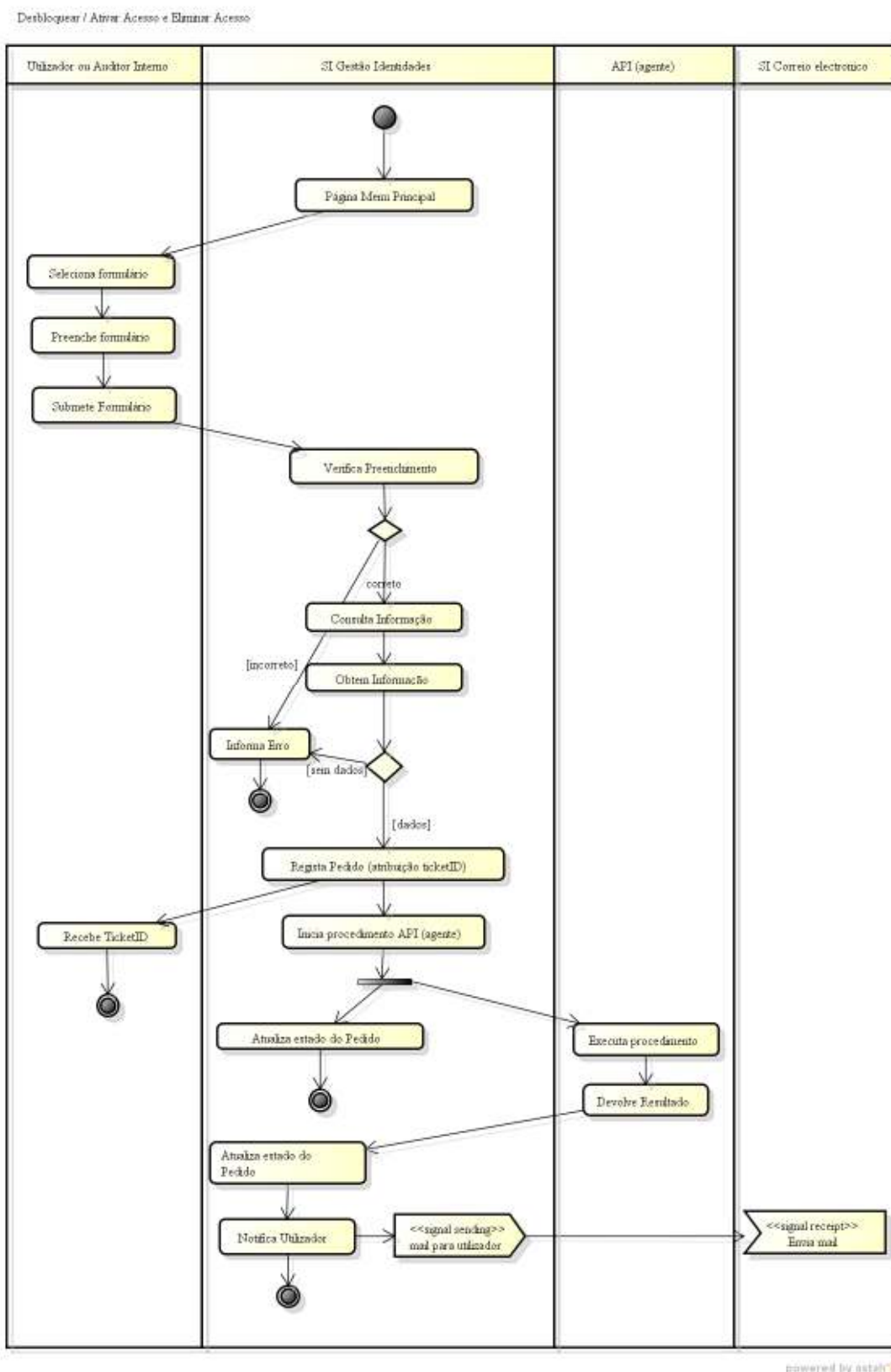


Fig. 24 – Diagrama de actividades: Desbloquear / Activar Acesso e Eliminar Acesso

As actividades Consultar Acessos / Perfil, Consultar Pedido e Eliminar Acesso apresentam-se igualmente como actividades e operações similares pelo que foram representadas no mesmo diagrama (figura 25).

Os fluxos das actividades e das operações não são complicadas, considerando que as respostas são baseadas nas informações armazenadas no sistema SI Gestão de Identidades, sendo no final disponibilizado aos actores a possibilidade de duas formas de recolha de resultados: Visualizar a Informação e Exportar Informação.

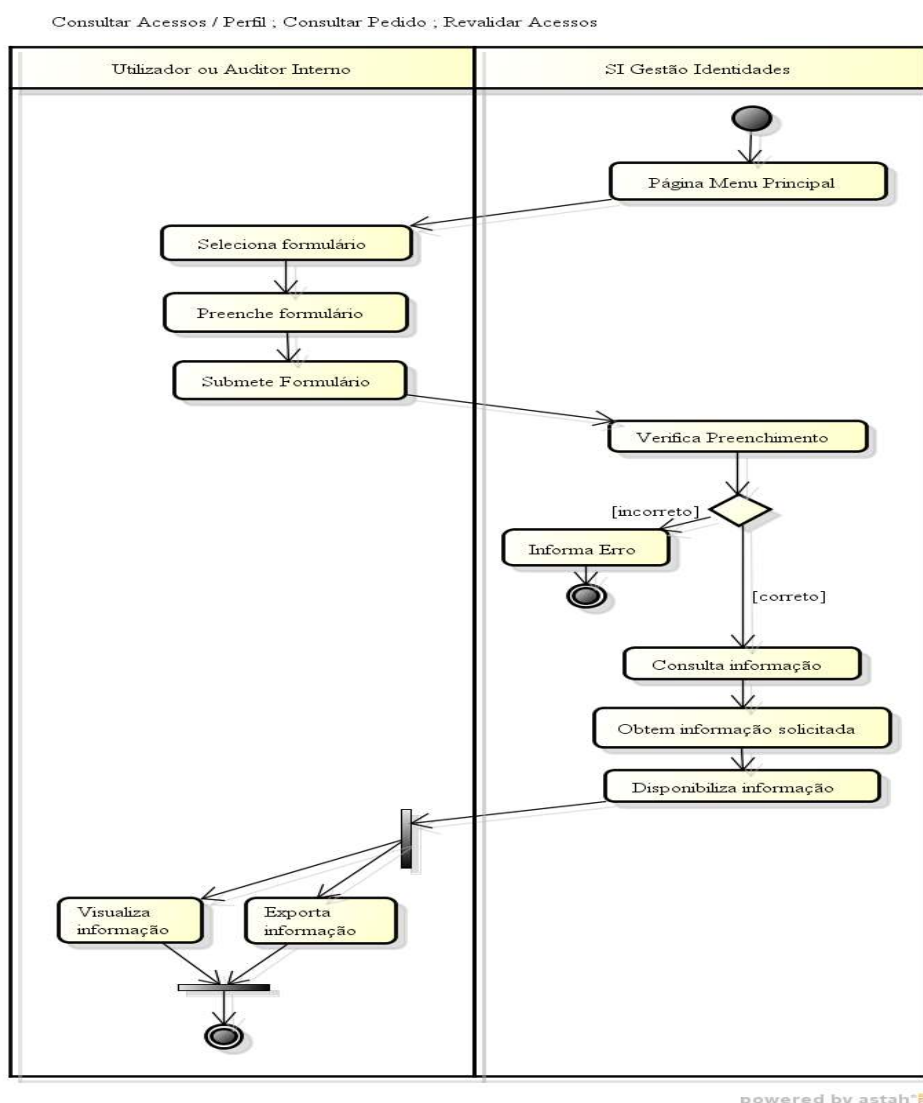


Fig. 25 – Diagrama de actividades: Consultar Acessos / Perfil, Consultar Pedidos e Revalidar Acessos

3.2.8. Diagramas de componentes (*Component Diagram*)

Um diagrama de componentes permite modular a arquitectura de um sistema na perspectiva dos componentes de *software*, (como por exemplo ficheiros código de fonte, executáveis, tabelas de dados, etc.) relacionando as suas múltiplas dependências.

Considerando a perspectiva *software*, foram identificados na figura 26 os componentes necessários considerando o módulo principal Solicitar Acesso.

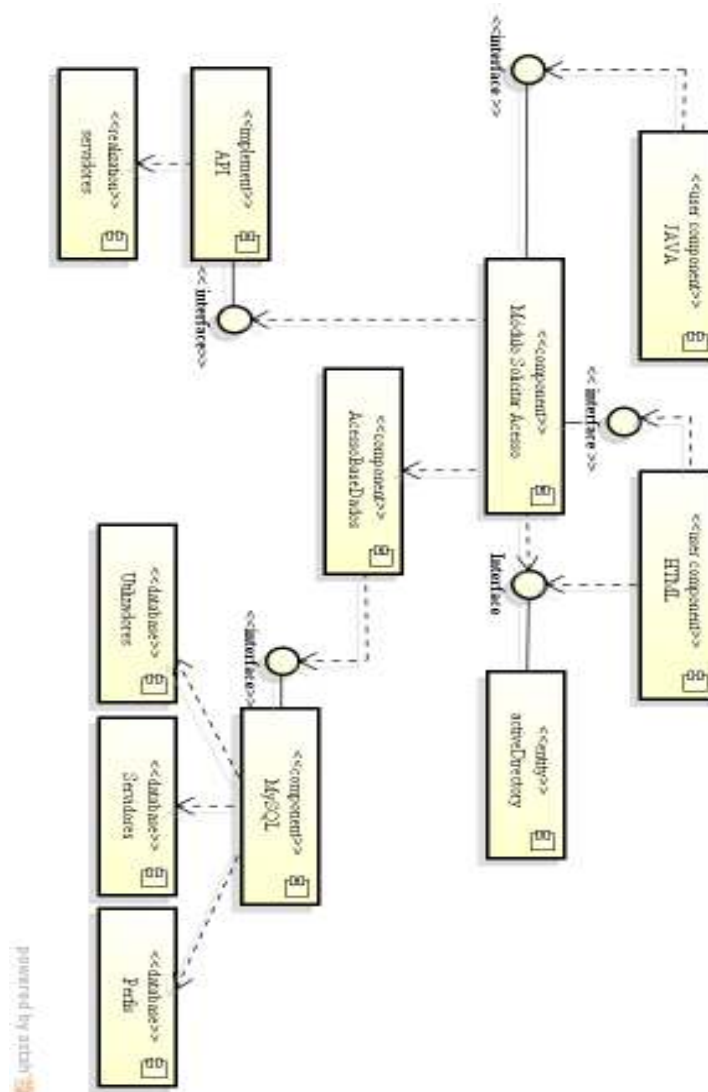


Fig. 26 – Diagrama de componentes: S.I. Gestão de Identidades

Este exercício foi efectuado para os restantes módulos (Consultar Acessos/Perfil, Desbloquear/Activar Acesso, Alterar Perfil, Autorizar Pedido, Revalidar Acessos, Eliminar Acesso, Validar Acesso, e Alterar Perfil), no entanto não foram identificados a necessidade de novos componentes, sendo o diagrama apresentado suficiente na consideração dos mesmos.

3.2.9. Diagramas de instalação (*Deployment Diagram*)

Um diagrama de instalação permite por sua vez modular a arquitectura de um sistema na perspectiva dos componentes de *hardware*, (como por exemplo computadores, cablagem, routers, etc.) relacionando as suas dependências de comunicação.

Considerando o diagrama de componentes acima apresentado, foi considerado a seguinte estrutura em termos de *hardware*, (conforme desenhado na figura 27):

O sistema é suportado por duas máquinas:

- a primeira (*Servidor HTTP*) suporta os componentes aplicativos
- e a segunda (*Servidor Base de Dados*) suporta o sistema de base de dados.

O sistema providencia uma interface *Web* (*Web Browser*) aos utilizadores a uma página de HTML², suportado tecnologicamente sobre um sistema de base de dados MySQL (onde se encontra alojada as bases de dados), por *APIs*³ desenvolvidas

² *HyperText Markup Language* (HTML) é uma linguagem de marcação (sendo uma linguagem de marcação um sistema de anotação de um texto de modo que ele seja sintaticamente distinguível) utilizada para produzir páginas na Web.

³ *Application Programming Interface* (API) definem-se como especificações baseadas em código-fonte destinadas a serem usado como interfaces em componentes de *software* para comunicar uns com os outros. As APIs podem incluir especificações de rotinas, estruturas de dados, classes de objectos e variáveis.

internamente e pelo sistema de permissões de acesso implementado (baseado na *Active Directory* e *Exchange* (correio electrónico)).

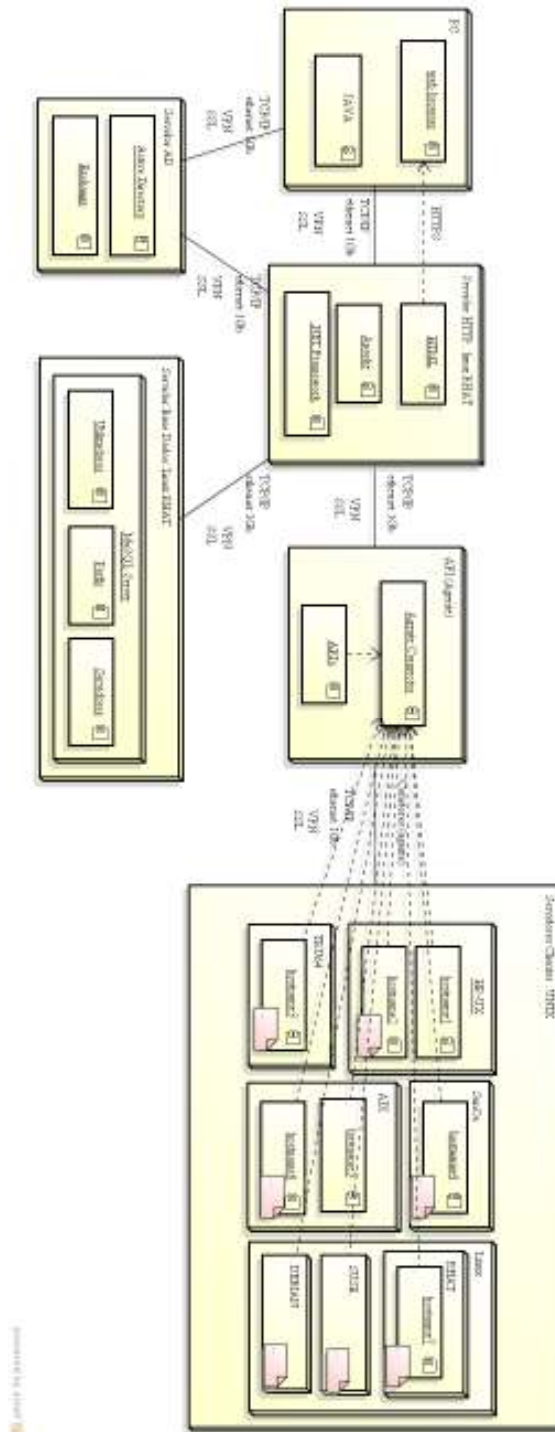


Fig. 27 – Diagrama de instalação: S.I. Gestão de Identidades

Através de um agente (API (agente)) instalado nos sistemas clientes (*hostname1, hostname2, hostname...*) é estabelecido a ligação e o envio da informação. Por sua vez, os sistemas clientes, independentemente da plataforma UNIX (HP-UX, TRU64, SunOs, AIX e LINUX), devem ter instalados o respectivo agente conector. O *software* (API's) a desenvolver deverá considerar os diferentes sistemas operativos e integrar a componente aplicacional com os servidores em questão, através do agente, que servirá de conector.

Os sistemas comunicam entre si de acordo com os protocolos seguros padronizados. Os protocolos são regras de comunicação que controlam e permitem estabelecer conexões, comunicações ou transferência de dados entre os diferentes sistemas computacionais. Para que os sistemas consigam comunicar é necessário que utilizem o mesmo protocolo. O protocolo TCP/IP [Loshin, Pete. (2003)] foi criado para que os equipamentos consigam "falar" entre si através da rede utilizando regras convencionadas que definem a sintaxe, semântica e sincronização da comunicação. Os protocolos podem ser implementados por *hardware* (nível mais baixo), *software* ou por combinação dos dois.

A comunicação entre os computadores clientes (PCs dos utilizadores) e a aplicação será efectuada considerando as configurações *end-to-end* de uma VPN⁴ entre os sistemas. A comunicação com as páginas de HTML fazem-se com recurso ao protocolo HTTPS, baseado na implementação do protocolo HTTP⁵ mas considerando uma camada adicional de segurança que utiliza o protocolo SSL⁶. Essa camada adicional (protocolo

⁴ As *Virtual Protocol Network* (VPNs) são conexões ponto a ponto em redes privadas ou públicas (exemplo a Internet). Um cliente VPN usa protocolos especiais baseados em TCP/IP, denominados protocolos de encapsulamento, para realizar chamadas virtuais a portas virtuais num servidor VPN. Geralmente a acesso e a troca de dados somente é permitido a utilizadores e/ou redes que façam parte da mesma comunidade de interesse.

⁵ *HyperText Transfer Protocol* (HTTP) é um protocolo de comunicação utilizado para sistemas de informação de hipermedia distribuídos e colaborativos. Normalmente, este protocolo utiliza a porta 80 e é usado para a comunicação de sites Web, comunicando na linguagem HTML.

⁶ O *Secure Socket Layer* (SSL) é um protocolo usado que permite estabelecer um acesso seguro ao encriptar toda a transmissão entre clientes e servidores.

SSL) possibilita que a informação seja transmitida através de conexões criptográficas⁷, permitindo que a autenticidade do servidor e do cliente seja verificada através de certificados digitais. A comunicação entre o agente conector e os sistemas UNIX será realizada por SSH⁸, de forma a executar comandos ou alterações remotas, possibilitando igualmente a conexão criptógrafa entre os sistemas.

Com recurso as estes protocolos e à criptografia usada internamente pelos mesmos garantem-se os princípios de segurança, nomeadamente o da **confidencialidade** da informação, em que só os destinatários autorizados são capazes de extraírem o conteúdo das mensagens da sua forma cifrada, e o princípio de **integridade** da informação, relacionados com a determinação se a mensagem foi alterada durante a transmissão.

⁷ A criptografia são princípios e técnicas sobre as quais os dados podem ser transformados da sua forma original para outra ilegível, de maneira que apenas possam ser conhecidos pelo seu destinatário (detentor da "chave secreta"), o que torna difícil que seja lido por alguém não autorizado. Uma chave criptográfica é um valor secreto que modifica um algoritmo de encriptação.

⁸ *Secure Shell* (SSH) é simultaneamente um programa de computador e protocolo cliente-servidor usado para permitir a comunicação entre computadores ligados numa rede. Possui as mesmas funcionalidades do TELNET, mas com a vantagem de estabelecer conexões criptografada entre o cliente e o servidor.

Conclusão

Estabelecer o equilíbrio entre custo, risco e flexibilidade, suportado por sistemas robustos é essencial numa organização que tenha de ter o risco técnico adequadamente medido e contido sem comprometer a qualidade e a sua agilidade operacional.

A segurança informática que hoje em dia está na base da segurança tecnológica, passa também pela implementação de meios automáticos transversais a toda a organização e que permitam gerir, com base numa única estrutura que cobre os vários *layers* existentes, sejam eles, sistemas, pessoas, identificação dos utilizadores na organização, palavras-chave, autorizações, responsáveis, etc.

Este trabalho surgiu como resposta a um desafio que passa em primeiro lugar pela racionalização de toda a infra-estrutura, seguida da utilização de ferramentas que permitam isolar a complexidade e automatizar a gestão dos utilizadores, tendo sempre como fim suportar os objectivos do negócio, melhorar o acesso dos utilizadores às aplicações, reduzir os custos de formação e de suporte, facilitar a administração dos sistemas e tornar os sistemas de informação mais flexíveis e dinâmicos, capazes de responderem de forma ágil a essas necessidades.

Nesse sentido, foi elaborada a construção de uma solução de Gestão de Identidades interna para ambientes UNIX com o objectivo de simplificar o processo e tornar o acesso e a gestão mais fácil e inteligente, quer do ponto de vista dos utilizadores, quer em termos da redução de complexidade e custos de exploração.

Através da criação deste portal é possível colocar à disposição do utilizador final ferramentas que permitem que os mesmos possam criar e gerir os seus acessos (como contas de utilizadores e palavras-chave), tomarem conhecimento da organização ao lhes ser apresentado, independentemente da infra-estrutura técnica, as diferentes opções que existem na organização (exemplo dos perfis), não descurando as validações de segurança necessárias a estes processos (exemplo para as situações que requerem aprovação).

Durante a elaboração do trabalho foram adquiridos conhecimentos e aprendidas algumas lições, nomeadamente:

- ✚ Que é essencial ter em consideração a cultura organizacional, as mudanças e principalmente os agentes de mudança;
- ✚ Fazer um planeamento correcto, considerando as diferentes etapas, tendo em vista que o processo é longo e complexo;
- ✚ Apresentar e divulgar internamente as mudanças, envolvendo as pessoas e garantindo a sua cooperação;
- ✚ Apresentar resultados parciais de forma a justificar o esforço envolvido;
- ✚ Definir os níveis de segurança, controlo de dados e informações adequadas às actividades.
- ✚ Analisar a informação existente e integrar também essa informação para evitar situações de conflito e, ou de erro.

Sendo um trabalho introdutório ao desenvolvimento de um sistema de Gestão de Identidades para ambientes UNIX, foi necessário efectuar o enquadramento na perspectiva dos sistemas envolvidos e do problema em si. Foram identificadas e seleccionadas algumas das alternativas presentes no mercado, no entanto, considerando que as mesmas não respondiam à solução pretendida para o problema, foi desenvolvido este projecto. Para além da análise do sistema, foi efectuado a identificação detalhada das funcionalidades do mesmo, nomeadamente o levantamento de requisitos e a respectiva descrição e especificação do sistema. Foi igualmente realizado a definição detalhada da arquitectura global da solução, com a apresentação dos módulos, tabelas, interfaces, sistemas envolvidos.

Como trabalhos futuros refiro desde já componente de desenvolvimento, tarefa na qual deverá ser realizada a programação dos diversos componentes do sistema, os testes de integração, onde o sistema será verificado com o objectivo de obter a aceitação e respectiva entrada em produção. Por último há que considerar a manutenção, relacionado sobretudo com o tempo de vida útil do sistema e com a entrada em funcionamento do mesmo.

Bibliografia

21 CFR Part 11 (1999-2009). Disponível on-line em: <http://www.21cfrpart11.com/>
<http://www.21cfrpart11.com/pages/library/index.htm>. Último acesso em 07-02-2012.

Assembly language (2011). Disponível on-line em:
http://en.wikipedia.org/wiki/Assembly_language. Último acesso em 08-12-2011.

Astah (2011). Disponível on-line em: <http://astah.net/editions/community>. Último acesso em 07-02-2012.

Bishop, M. (2002). *Computer Security: Art and Science*. Boston: Addison-Wesley Professional.

C (programming language) (2011). Disponível on-line em:
[http://en.wikipedia.org/wiki/C_\(programming_language\)](http://en.wikipedia.org/wiki/C_(programming_language)). Último acesso em 08-12-2011.

COBIT (2011). *COBIT Framework for IT Governance and Control*. Disponível on-line em: <http://www.isaca.org/Knowledge-Center/COBIT/Pages/Overview.aspx>. Último acesso em 26-11-2011.

Carneiro, A. (2009). *Auditoria e Controlo de Sistemas de Informação*. Lisboa: FCA – Editora Informática, Lda.

Gramm-Leach-Bliley (2001). Disponível on-line em: <http://business.ftc.gov/privacy-and-security/gramm-leach-bliley-act>. Último acesso em 07-02-2012.

HIPAA (2011). *Health Insurance Portability and Accountability Act of 1996 (HIPAA) Privacy and Security Rules*. Disponível on-line em: <http://www.hhs.gov/ocr/privacy/hipaa/understanding/index.html>. Último acesso em 07-02-2012.

HSPD-12 (2012). *Homeland Security Presidential Directive 12 (HSPD-12)*. Disponível on-line em: <http://hspd12.usda.gov/about.html>. Último acesso em 07-02-2012.

ISO (2011). *International Standards for Business, Government and Society*. Disponível on-line em: <http://www.iso.org/iso/home.htm>. Último acesso em 26-11-2011.

ITIL (2007). Disponível on-line em: <http://www.ital-officialsite.com/>. Último acesso em 26-11-2011.

Loshin, Pete. (2003). *TCP/IP Clearly Explained, FOURTH EDITION*. California, San Francisco: Morgan Kaufmann Publishers.

Love, P., Merlino, J., Reed, J.C., Zimmerman, C. e Weinstein, P. (2005). *Beginning Unix®*. Indianapolis, Indiana: Wiley Publishing, Inc

Mamede, H. S. (2006). *Segurança Informática nas Organizações*. Lisboa: FCA – Editora Informática, Lda.

OASIS (2012). *Organization for the Advancement of Structured Information Standards (OASIS)*. Disponível on-line em: <http://www.oasis-open.org/org>. Último acesso em 07-02-2012.

OMG (1997-2012). *Unified Modeling Language*. Disponível on-line em: <http://www.uml.org/>. Último acesso em 07-02-2012.

Pfitzmann, A. e M. Hansen, (2010). “*A terminology for talking about privacy by data minimization: Anonymity, Unlinkability, Undetectability, Unobservability, Pseudonymity, and Identity Management*”. Disponível on-line em: http://dud.inf.tu-dresden.de/literatur/Anon_Terminology_v0.34.pdf. Último acesso em 07-02-2012.

Saltzer, J.H. e Schroedder, M.D. (1974). *The Protection of Information in Computer Systems*. Disponível on-line em: <http://www.cs.virginia.edu/~evans/cs551/saltzer/>. Último acesso:14-03-2012.

Silva, A. e C.Videira (2005). *UML, Metodologias e ferramentas CASE*. Lisboa: Centro Atlântico, Lda

SOX (2006). *A Guide To The Sarbanes-Oxley Act*. Disponível on-line em: <http://www.soxlaw.com/>. Último acesso em 26-11-2011.

UNIX System (1995). *The Unix system*. Disponível on-line em: <http://www.unix.org/>. Último acesso em 24-01-2012.

Anexos

Portal SI Gestão de Identidades

Esta secção pretende apresentar o portal SI Gestão de Identidades que será disponibilizado aos utilizadores.

Considerando a especificação funcional descrita no projecto foram desenhados e elaborados os ecrãs que a seguir se apresentam.

Página de Login

Página de Login - Página Inicial de acesso ao portal.

Acesso restrito a pessoas autorizadas

Gestão de Identidades

Utilizador AD

Palavra-chave

Por favor preencha todos os campos e de seguida clique em aceder.

Em caso de dificuldade enviar mail para gestacessos@gmail.com

Fig. 28 – S.I. Gestão de Identidades: Página Login

Página Menu Principal

Página Menu Principal – Página de acesso aos diferentes formulários. É constituído pelo menu de opções, sendo que algumas (Revalidar Acessos e Autorizar Pedido) são de acesso restrito para o funcionário com perfil de Auditor Interno e para os funcionários com perfil Responsável / Autorizador, respectivamente.

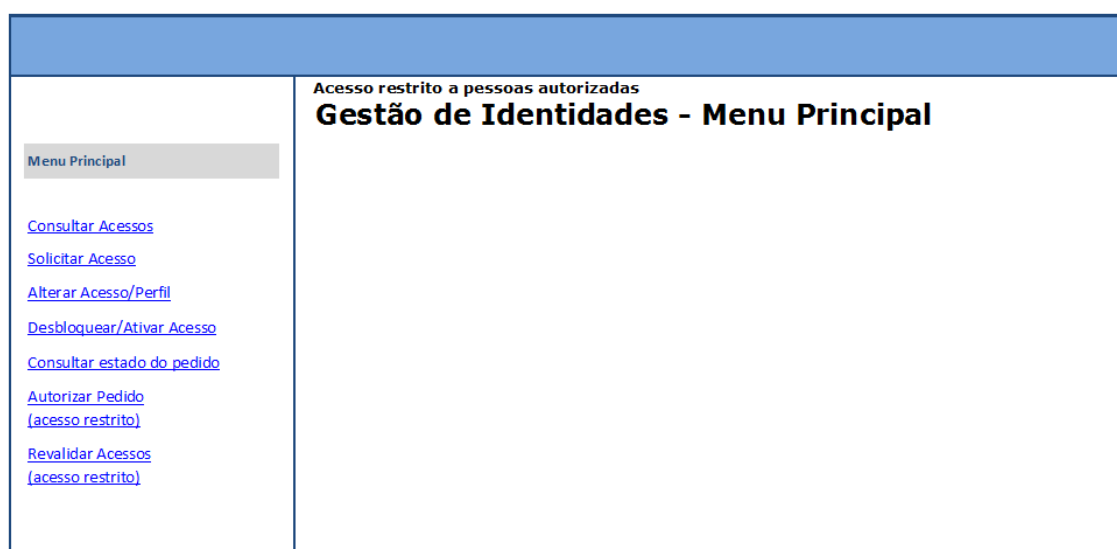


Fig. 29 – S.I. Gestão de Identidades: Página Menu Principal

Menu Consultar Acessos

Página de Consultar Acessos - Página que permite aos utilizadores validarem se tem uma conta de utilizador e, em que sistemas.

Os campos sombreados são campos cujo preenchimento é baseado no *login AD* e não são alteráveis pelo utilizador.

The screenshot shows a web interface for identity management. At the top, there's a blue header. Below it, a navigation sidebar on the left contains links: 'Consultar Acessos', 'Consultar Acessos', 'Solicitar Acesso', 'Alterar Acesso/Perfil', 'Desbloquear/Ativar Acesso', 'Consultar estado do pedido', 'Autorizar Pedido (acesso restrito)', and 'Revalidar Acessos (acesso restrito)'. The main content area is titled 'Acesso restrito a pessoas autorizadas' and 'Gestão de Identidades - Consultar Acessos'. It has a section for 'Identificação do Utilizador' with four input fields: 'Username', 'Login AD', 'Nome', and 'E-mail'. Below this is a 'Servidores' section with a large empty box and two buttons: 'Listar' and 'Exportar'.

Fig. 30 – S.I. Gestão de Identidades: Página Consultar Acessos / Perfil


Menu Solicitar Acesso

Página Solicitar Acesso – é talvez uma das páginas principais deste projecto, uma vez que procura resolver o problema (bastante frequente nas auditorias e nas evidências que necessitam de ser produzidas, que é:) o registo de quem pede, quando, o quê, em que servidores e autorizado por quem.

Através do SI Gestão de Identidades é possível produzir essa informação e apresentá-la nas situações de auditoria, considerando a base de dados que vai estar no suporte à infra-estrutura.

The screenshot shows a web application interface for requesting access. The main heading is "Gestão de Identidades - Solicitar Acesso". The interface is divided into several sections: 1. "Identificação do utilizador" (User Identification) with input fields for Username, Login AD, Nome (Name), and E-mail. 2. "Identificação Perfil" (Profile Identification) with a dropdown menu for "Perfil" currently set to "Selecionar". 3. "Servidores" (Servers) section containing two empty list boxes and four buttons: "Adicionar >>", "<< Remover", "Adicionar todos >>", and ">> Remover todos". 4. A "Submeter Pedido" (Submit Request) button at the bottom right. A left sidebar contains navigation links: "Menu Solicitar Acesso", "Consultar Acessos", "Solicitar Acesso", "Alterar Acesso/Perfil", "Desbloquear/Ativar Acesso", "Consultar estado do pedido", "Autorizar Perfil (Acesso restrito)", and "Revogar Acesso (Acesso restrito)".

Fig. 31 – S.I. Gestão de Identidades: Página Solicitar Acesso

Ainda como apoio aos utilizadores e, considerando a existência de novos funcionários / utilizadores dos sistemas Unix é disponibilizado a informação do campo *username* e que pode ser visualizado ao clicarem no ícone .


-  Conta de utilizador pessoal e intransmissível, associada ao *Login AD*, que permite a autenticação e autorização de acesso a um servidor ou a serviços de rede.

Fig. 32 – S.I. Gestão de Identidades: campo informativo

Menu Alterar Acesso / Perfil

Página Alterar Acesso / Perfil – Página que permite que o utilizador submeta pedidos de alteração do perfil do utilizador, considerando as alterações de funções dentro das empresas.

Acesso restrito a pessoas autorizadas
Gestão de Identidades - Alterar Perfil

Identificação do Utilizador

Username
Login AD
Nome
E-mail

Servidor

Servidor

Alterar Perfil

Perfil anterior
Perfil

Menu Alterar Acesso/Perfil

- [Consultar Acessos](#)
- [Solicitar Acesso](#)
- [Alterar Acesso/Perfil](#)
- [Desbloquear/Ativar Acesso](#)
- [Consultar estado do pedido](#)
- [Autorizar Pedido \(acesso restrito\)](#)
- [Revalidar Acessos \(acesso restrito\)](#)

Fig. 33 – S.I. Gestão de Identidades: Página Alterar Perfil

Menu Desbloquear / Activar Acesso

Página de Desbloquear / Activar Acessos - Página que permite que o próprio utilizador possa revalidar o seu acesso no sistema seleccionado (pressupõe que a conta existe no sistema. O SI Gestão de Identidade validará essa informação).

Acesso restrito a pessoas autorizadas
Gestão de Identidades - Desbloquear / Ativar Acesso

Identificação do Utilizador

Username
Login AD
Nome
E-mail

Desbloquear / Ativar Acesso

Ativação do Utilizador
 Alterar palavra-chave

Servidor

Servidor

Menu Desbloquear/Ativar Acesso

[Consultar Acessos](#)
[Solicitar Acesso](#)
[Alterar Acesso/Perfil](#)
[Desbloquear/Ativar Acesso](#)
[Consultar estado do pedido](#)
[Autorizar Pedido \(acesso restrito\)](#)
[Revalidar Acessos \(acesso restrito\)](#)

Fig. 34 – S.I. Gestão de Identidades: Página Desbloquear / Activar Acesso

Menu Consultar Estado do Pedido

Página para Consultar Estado do Pedido - Página que permite verificar o estado dos pedidos submetidos.

Acesso restrito a pessoas autorizadas

Gestão de Identidades-Consultar Estado do Pedido

Identificação do Pedido

Pesquisa por ID Pedido
 por Login AD
 por Data

Estado do Pedido

Submeter Pedido

Fig. 35 – S.I. Gestão de Identidades: Página Consultar Estado do Pedido

Menu Autorizar Pedido

Página Autorizar Pedido - Página restrita ao funcionário com perfil Responsável / Autorizador. Ao submeter o parecer sobre o pedido (“Autorizar” / Não Autorizar”) , o Responsável / Autorizador pode registar observações que entenda necessárias, sendo no entanto este campo, somente de comentário para efeitos de auditoria, isto é, não pressupõe actividade extra ao pedido. Por exemplo, no caso de rejeitar o pedido é conveniente colocar o motivo.

Acesso restrito a pessoas autorizadas

Gestão de Identidades - Autorizar Pedido

Identificação do Pedido

ID Pedido

Username

Perfil

Login AD

Nome

E-mail

Parecer

Autorizar

Não Autorizar

Observações

Fig. 36 – S.I. Gestão de Identidades: Página Autorizar Pedido

Menu Revalidar Acessos

Página Revalidar Acessos - Página restrita ao funcionário com perfil Auditor Interno. Possibilidade de listar a informação por *username* (campo por que um utilizador é conhecido no sistema Unix), por *Login AD* (conta pelo que um utilizador é conhecido na empresa) ou por servidor.

Acesso restrito a pessoas autorizadas

Gestão de Identidades - Revalidar Acessos

Identificação do Utilizador / Servidor

Listar por Username

por Login AD

por Servidor

Servidores

--

Listar

Exportar

Fig. 37 – S.I. Gestão de Identidades: Página Revalidar Acessos

Menu Eliminar Acessos

Página Eliminar Acessos - Página restrita ao funcionário com perfil Auditor Interno.

Poderá seleccionar:

- servidor a servidor, quando a eliminação é, por exemplo somente para descontinuação de determinado perfil,
- ou todos, quando a eliminação é em todos os servidores da empresa (exemplo para os casos em que o utilizador sai da empresa).

Acesso restrito a pessoas autorizadas

Gestão de Identidades - Eliminar Acessos

Identificação do Utilizador

Username

Servidores

Adicionar >>

<< Remover

Adicionar todos >>

>> Remover todos

Submeter Pedido

Fig. 38 – S.I. Gestão de Identidades: Página Eliminar Acessos

ⁱ UNIX é uma marca registrada licenciada através da X / Open Company (The Open Group).

ⁱⁱ *Microsoft, Active Directory*, o logotipo do *Office, SharePoint, Visual Studio, Windows*, e o logótipo *Windows* são marcas registradas ou marcas comerciais da Microsoft Corporation.

ⁱⁱⁱ *Oracle e Java* são marcas registradas da Oracle e/ou de suas associadas.

^{iv} *IBM, IBMlogo, ibm.com e Tivoli* são marcas registradas da International Business Machines Corporation (IBM).