FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Confidencialidade e Segurança da Informação

Bruno Tiago Lopes Conceição

Mestrado Integrado em Engenharia Eletrotécnica e de Computadores

Orientador: Professor Gil Manuel Gonçalves

Proponente: Engenheira Telma Salgueiro - IPBRICK SA

Janeiro de 2019

Resumo

O mundo empresarial cada vez mais depende de informações digitais para gerir os seus negócios, sendo que essas informações podem ser consideradas o bem mais importante a proteger numa organização.

Esta dissertação foi realizada em ambiente empresarial, na IPBRICK SA, e foi dividida em duas partes, sendo a primeira mais associada ao tema da confidencialidade, no Contacts, e a segunda relativa à segurança da informação, no iPortalDoc.

Numa primeira fase, foi realizado um estudo teórico sobre *Active Directory*, tendo como objetivo entender o conceito de LDAP. Numa fase posterior, foi feita uma investigação acerca de *Data-centric security*, apresentando alguns dos métodos mais adequados para a implementação de uma segurança centrada nos dados, numa aplicação *web*.

Após ter sido feita a caracterização detalhada dos problemas que motivaram a realização desta dissertação, foram descritas as arquiteturas do sistema e da aplicação, para o Contacts e para o iPortalDoc.

A primeira parte da dissertação, relativa à implementação, consistiu no desenvolvimento de duas melhorias associadas ás permissões dos utilizadores e à confidencialidade de entidades e contactos, e na elaboração de uma solução para um problema relacionado com a transferência de permissões para o servidor LDAP, a partir do Contacts. A fase de implementação relativa ao primeiro problema, consistiu no desenvolvimento de uma melhoria no Contacts, que permitiu que os tipos de entidade que tivessem todas as suas entidades sem utilizadores associados, que se tornassem automaticamente públicas a todos os utilizadores da aplicação. A implementação da segunda melhoria possibilitou que o Contacts passasse a ter uma nova ferramenta de pesquisa de utilizadores, que é capaz de os filtrar por entidade ou tipo de entidade; o desenvolvimento desta nova funcionalidade leva a que haja uma drástica redução do tempo que habitualmente era necessário, diminuindo também a probabilidade de errar ao selecionar os utilizadores que devem ter acesso às entidades e contactos. Relativamente ao último problema da primeira parte, foi elaborada uma solução e feita a implementação do lado do LDAP, fazendo com que o LDAP passe a gerir as permissões (definidas no Contacts) de cada utilizador; o desenvolvimento desta solução permite corrigir um problema que consistia no facto de, até ao momento, por não serem transmitidas as permissões de cada utilizador para o servidor LDAP, as aplicações como o IPBRICK.MAIL, disponibilizassem as informações sobre todas as entidades e contactos, para todos os utilizadores, comprometendo a sua confidencialidade.

Hoje em dia, não basta haver uma segurança apenas ao nível dos servidores ou de aplicações de uma empresa. É necessário que haja uma segurança baseada nos dados (*Data-centric security*). Assim, o objetivo da segunda parte da dissertação foi implementar funcionalidades chave de *Data-centric security*, na aplicação de gestão de documentos e processos, o iPortalDoc. Primeiramente, foi analisado um método de prevenção de perda de dados, que teria as funcionalidades pretendidas, mas que, numa fase mais avançada, foi entendido que não seria possível avançar com a sua implementação na empresa. Depois, foi explorado um método de proteção de dados, tendo

sido analisado e verificado, mas que também não pôde ser avançada a sua implementação, desta vez pela versão antiga do sistema operativo disponibilizado. Embora não tenha sido concretizada a implementação destes dois métodos, os resultados obtidos permitirão uma fácil adaptação a ser realizada em trabalho futuro.

Abstract

The business world is increasingly dependent on digital information to run its business, and that information can be considered the most important asset to protect in an organization.

This dissertation was conducted in a business environment, at IPBRICK SA, and was divided into two parts, the first one associated with the topic of confidentiality in Contacts, and the second one related to security of information, in iPortalDoc.

In a first phase, a theoretical study was carried out on Active Directory, aiming to understand the concept of LDAP. In the later phase, an investigation was made about Data-centric security, presenting some of the most appropriate methods for the implementation of data-centric security.

After the detailed characterization of the problems that motivated this dissertation, the system and application architectures for Contacts and iPortalDoc were described.

The first part of the dissertation was the implementation of two improvements regarding user permissions and the confidentiality of entities and contacts, and the elaboration of a solution to a problem related to the transfer of permissions to the LDAP server through Contacts. The implementation phase related to the first problem consisted in the development of an improvement in Contacts, which allowed that entity types that had all their entities without associated users would become automatically public to all the users of the application. The implementation of the second improvement enabled Contacts to have a new user search tool, which is able to filter them by entity or entity type; the development of this new functionality leads to a drastic reduction of the time that was usually necessary, also reducing the probability of error when selecting the users who must have access to the entities and contacts. Regarding the last problem of the first part, a solution was elaborated and the implementation of the LDAP side was made, causing LDAP to manage the permissions (defined in the Contacts) of each user; the development of this solution makes it possible to correct a problem which was that, so far, because of each user's permissions to the LDAP server were not transmitted, applications such as IPBRICK.MAIL would provide information about all entities and contacts, for all users, compromising their confidentiality.

Nowadays, it is no longer enough to have security only at the level of a company's servers or applications. Data-centric security is required. Thus, the purpose of the second part of the dissertation was to implement key data-centric security features in the document and process management application, iPortalDoc. Firstly, a method of preventing data loss was analyzed, which would have the desired functionalities, but at a later stage it was understood that it would not be possible to proceed with its implementation in the company. Afterwards, a method of data protection was explored, having been analyzed and verified, but couldn't also be advanced its implementation, this time because of the old version of the operating system available. Although the implementation of these two methods has not been implemented, the results obtained will allow an easy adaptation to be carried out in future work.

Agradecimentos

Quero começar por agradecer ao orientador, Professor Gil Manuel Gonçalves, pela disponibilidade, pelo apoio e pelas dicas que me ajudaram a elaborar a dissertação.

Aos meus pais e irmão, por todo o apoio durante a minha vida.

À minha namorada, Ana Cunha, por todo o apoio, ajuda, motivação e por fazer parte do meu sucesso.

À IPBRICK SA, pela oportunidade de desenvolver esta dissertação, bem como a todos os seus colaboradores que me ajudaram durante esta fase, em especial à Eng. Telma Salgueiro, pela oportunidade, e ao André Morais e ao João Castro que sempre se prestaram a ajudar-me.

Aos meus amigos e colegas, que de uma ou outra forma me ajudaram, não só neste trabalho, mas durante todo o curso.

Bruno Tiago Lopes Conceição

"The definition of insanity is
doing the same thing over and over again
and expecting different results."

Albert Einstein

Conteúdo

Lista de Figuras

Lista de Tabelas

Abreviaturas e Símbolos

ACL	*Access Control List*
AD	*Active Directory*
AES	*Advanced Encryption Standard*
DDM	*Dynamic Data Masking*
DHCP	*Dynamic Host Configuration Protocol*
DLP	*Data Loss Prevention*
DNS	*Domain Name System*
FTP	*File Transfer Protocol*
GPG	*Gnu Privacy Guard*
GUID	*Globally Unique Identifier*
HTML	*HyperText Markup Language*
HTTP	*Hypertext Transfer Protocol*
HTTPS	*Hyper Text Transfer Protocol Secure*
IT	*Information Technology*
JSON	*JavaScript Object Notation*
LDAP	*Lightweight Directory Access Protocol*
LDIF	*Lightweight Directory Import Format*
NOS	*Network Operating System*
OS	*Operating System*
OU	*Organizational Unit*
PCI-DSS	*Payment Card Industry Data Security Standard*
PGP	*Pretty Good Privacy*
PHP	*PHP: Hypertext Preprocessor*
RBAC	*Role-Based Access Control*
RDN	*Relative Distinguished Name*
RGPD	Regulamento Geral de Proteção de Dados
SMS	*Short Message Service*
SMTP	*Simple Mail Transfer Protocol*
SQL	*Structured Query Language*
TCP/IP	*Transmission Control Protocol/Internet Protocol*
UUID	*Universally Unique Identifier*
VPN	*Virtual Private Network*
XML	*Extensible Markup Language*

Capítulo 1

Introdução

1.1 Contexto

Atualmente, o mundo empresarial não pode cingir-se apenas a vender um produto. É necessário proteger o cliente, a empresa e o produto que é comercializado. Com o passar dos anos, melhorias têm sido feitas no que diz respeito à segurança, mas ao mesmo tempo, os ataques informáticos são cada vez mais evoluídos. Investir apenas na segurança de redes ou servidores de uma empresa, pode não ser suficiente para proteger dados digitais confidenciais. É essencial haver uma segurança centrada em dados (*Data-centric security*).

Data-centric security é uma abordagem de segurança que enfatiza a segurança dos dados em vez da segurança em redes, servidores ou aplicações. A segurança centrada em dados está a evoluir à medida que as organizações dependem cada vez mais de informações digitais para gerir os seus negócios [7]. Com base neste tipo de segurança, é possível restringir o acesso aos dados. Ou seja, é possível definir as permissões que um utilizador tem sobre determinados dados, em que podem ser, por exemplo, permissões de visualização, edição, criação ou impressão. Desta forma, o utilizador, a quem lhe for concedido acesso ao documento e com a restrição de que apenas o pode visualizar, não irá conseguir copiar o conteúdo para outro documento nem conseguirá imprimi-lo. Numa simples frase, pode-se dizer que a proteção viaja com o documento, onde quer que vá.

A IPBRICK SA pretende melhorar a segurança dos dados que estão contidos na sua aplicação de gestão documental e de processos, o iPortalDoc. Esta aplicação *web* possibilita gerir todos os documentos de uma organização, permitindo introduzir, registar, classificar e encaminhar, bem como fazer o tratamento processual de todas as atividades de uma organização.

Com a entrada em vigor do Regulamento Geral de Proteção de Dados (RGPD), todas as organizações têm a obrigação de informar em que situações irão utilizar dados de clientes ou funcionários. Por esta razão, é essencial manter a confidencialidade das informações dos clientes.

A IPBRICK SA utiliza o Contacts para a gestão das entidades e contactos. Pelo facto do RGPD ter entrado em vigor em 2018, é necessário assegurar que a confidencialidade das entidades e contactos não é posta em causa, com a utilização desta aplicação *web*.

1.2 Motivação

No dia 25 de maio de 2018, entrou em vigor o Regulamento Geral de Proteção de Dados (RGPD). Esta lei de privacidade afeta todas as organizações que têm negócio com ou dentro da União Europeia.

O Regulamento Geral de Proteção de Dados permitirá que todos tenham um maior controlo sobre os seus dados pessoais. Se uma organização trabalhar com dados pessoais, terá que ter uma razão legal e válida para tal. Desta forma, não será possível que as empresas peçam dados para uma situação e os utilizem para outra [8].

Assim sendo, a IPBRICK SA pretende reforçar a confidencialidade dos dados contidos nas suas aplicações, e melhorar a segurança dos mesmos, nas aplicações de gestão de contactos e de gestão de documentos e processos.

1.3 Objetivos

Os principais objetivos desta dissertação são:

- Estudo e análise do protocolo LDAP e de abordagens *Data-centric security* a implementar em aplicações;
- Análise e implementação de melhorias relacionadas com a confidencialidade, no Contacts;
- Elaboração de uma solução para um problema associado à confidencialidade de entidades e contactos, relacionado com as permissões que o Contacts transmite para o servidor LDAP;
- Desenvolvimento e implementação de uma abordagem baseada em *Data-centric security*, no iPortalDoc;
- Verificação e validação das funcionalidades implementadas nas duas aplicações empresariais;

1.4 Estrutura da Dissertação

Este documento é composto por seis capítulos.

No presente capítulo, é feita uma introdução à dissertação, apresentando o seu contexto, a motivação e os objetivos propostos.

No capítulo 2, é descrito o estado da arte sobre *Active Directory*, mais especificamente sobre LDAP; é também feita uma análise relativa a *Data-centric security*, onde são apresentados todos os passos para uma implementação com sucesso num ambiente empresarial; em seguida, são apresentadas aplicações existentes no mercado baseadas neste tipo de segurança; por fim, são fornecidas bases teóricas acerca das tecnologias que irão ser utilizadas e é feita uma introdução ás aplicações (da IPBRICK SA) relevantes para esta dissertação.

O capítulo 3 é constituído pela definição geral do problema, onde é apresentado e detalhado o problema e o que é pretendido desenvolver neste projeto, em cada situação.

No capítulo 4, é apresentada a arquitetura do sistema e da aplicação do Contacts e do iPortal-Doc. Posteriormente, é elaborada e exibida a solução para cada problema do projeto, explicitado no capítulo 3.

No capítulo 5, é feita uma introdução ao *software* que foi utilizado durante a implementação; posteriormente, é demonstrada toda a implementação efetuada, são apresentados os resultados e é feita a validação em cada fase.

Finalmente, no capítulo 6, são apresentadas as principais conclusões e o trabalho futuro que poderá ser realizado.

Capítulo 2

Revisão Bibliográfica

Neste capítulo é feita uma introdução à necessidade de proteger os dados de uma organização, ao conceito de *Active Directory* e a *Data-centric security*. Também são demonstradas ferramentas já existentes, assim como alguns exemplos de aplicações que as utilizam. Para além disso, são ainda apresentadas algumas bases teóricas relativas ás tecnologias e ao software que irão ser utilizados durante a dissertação.

2.1 Regulamento Geral de Proteção de Dados

O regulamento geral de proteção de dados (RGPD) entrou em vigor em maio de 2018, permitindo aos cidadões e residentes na união europeia terem controlo sobre a sua informação pessoal.

O RGPD regula a forma como os dados pessoais do consumidor são armazenados, apagados, transferidos e utilizados. Isto acontece não só para as bases de dados, como também para sistemas *e-commerce*, páginas de redes sociais, servidores baseados em *cloud*, etc. De acordo com o regulamento, a definição de "dados pessoais"inclui todas as informações relacionadas a alguém que permitam identificar essa pessoa. Informações identificáveis podem ser um nome, uma foto, um endereço de e-mail ou detalhes duma conta bancária.

O RGPD define cinco alterações importantes na forma como as organizações recolhem e protegem os dados pessoais dos seus clientes [9]:

- Obtenção de consentimento – as organizações devem agora obter o consentimento para recolher dados pessoais;
- Direito ao esquecimento – os cidadãos ou residentes da união europeia podem solicitar que os seus dados pessoais sejam completamente removidos dos sistemas e registos de uma organização, devendo essa organização cumprir e apresentar a prova de conformidade;
- Transferência de dados – Da mesma forma, os utilizadores podem solicitar que uma organização transfira todos os seus dados para outra organização;

- Responsável pela proteção de dados – Organizações de um tamanho acima do definido no regulamento, são obrigadas a nomear um responsável pela proteção de dados para supervisionar a utilização de dados dos clientes;

- Notificação de violação de segurança – No caso de uma violação de segurança, as organizações devem notificar os cidadãos ou residentes da união europeia afetados no prazo de 72 horas após tomarem conhecimento da violação, e devem iniciar imediatamente a correção. Os afetados devem receber uma explicação do sucedido e podem iniciar uma ação legal contra a organização.

2.2 Segurança da Informação

Nem tudo o que não cria valor para um produto pode ser descartado. É necessário haver um investimento adequado na segurança de uma organização. Segundo [10], a segurança não pode ser ignorada mas, ao mesmo tempo, deve ser de baixo custo.

Embora a segurança seja um fator imprescindível nas empresas, as opiniões sobre o nível de segurança que é necessário, diferenciam de umas para as outras. Investir na segurança é visto pelas organizações como custos sem benefícios tangíveis para o negócio, e, portanto, esses custos devem ser minimizados. Para isso acontecer, é fundamental implementar uma segurança objetiva àquilo que é crucial proteger.

A segurança de informação está intrinsecamente relacionada com a prevenção de acesso não autorizado, uso, partilha, modificação e destruição de informação.

Os pilares da segurança de informação são:

- Confidencialidade – diz respeito à proteção de acesso não autorizado a documentos;

- Integridade – significa que um documento não foi modificado;

- Disponibilidade – os recursos e a infraestrutura deverão manter-se totalmente funcionais, em todas as situações;

- Não repúdio – nenhuma das partes pode negar o envio, recebimento ou o acesso aos dados, o que implica um grau de auditabilidade;

- Autenticidade – informa que a origem pode ser identificada.

2.3 *Active Directory*

O *Active Directory* (AD) é um sistema operativo de rede (Network Operating System - NOS) da Microsoft. Inicialmente desenvolvido sobre o Windows 2000, o AD evoluiu durante mais de uma década através de múltiplas versões do Windows.

O AD permite aos administradores gerirem eficazmente toda a informação da empresa, a partir dum repositório central que pode ser globalmente distribuído. Assim que a informação dos utilizadores e grupos, computadores e impressoras, e aplicações e serviços for adicionada ao *Active Directory*, esta poderá ser disponibilizada para uso, para toda a empresa. A estrutura da informação pode coincidir com a estrutura da organização, e os seus utilizadores podem consultar o *Active Directory* para encontrarem a localização de uma impressora ou de um endereço de e-mail de um colega. Com unidades organizacionais (OU), é possível delegar o controlo e a gestão dos dados da forma que mais convém para cada caso.

O NOS é descrito como um ambiente de rede onde vários tipos de recursos, tais como, contas de utilizadores, de grupos, e de computadores, são guardados num repositório central controlado por administradores e acessível a utilizadores finais. Normalmente, um ambiente NOS é composto por um ou mais servidores que fornecem serviços NOS, tais como autenticação, autorização, e manipulação de contas, por múltiplos utilizadores que tenham acesso a esses serviços.

O conceito de "domínio" foi introduzido no Windows NT, disponibilizando um modo de agrupar recursos baseado nos limites administrativos e de segurança. Os domínios NT eram estruturas limitadas a cerca de 40000 objetos (utilizadores, grupos e computadores). Para grandes organizações, esta restrição impunha limites superficiais no design da estrutura do domínio. Frequentemente, os domínios também eram limitados geograficamente pois a replicação dos dados entre controladores de domínio tinham um desempenho pobre em altas latências ou ligações de baixa largura de banda. Outro problema significativo era a delegação de administração, que tendia a ser uma questão de "tudo ou nada" a nível do domínio [11].

Havendo a necessidade de reestruturar o modelo NOS de modo a tornar em algo escalável e mais flexível, foi criado o *Lightweight Directory Access Protocol* (LDAP).

2.3.1 LDAP

LDAP é a tecnologia *standard* para um cliente aceder a um serviço de diretório.

Esta tecnologia funciona em qualquer tipo de rede TCP/IP e simplifica a implementação do acesso ao diretório através de métodos que minimizam os requisitos e a complexidade dos recursos. Também traz vantagens como uma rápida e avançada pesquisa, uma rápida resposta e uma visualização dos dados hierarquicamente [12].

Através deste protocolo, é possível organizar os recursos de rede de forma hierárquica, sendo que no topo se encontrará o diretório da raiz, de seguida a rede da empresa, depois o departamento e, por fim, o computador do funcionário e os recursos da rede compartilhados pelo mesmo. Na figura 2.1 encontra-se um exemplo de uma árvore de diretório, que demonstra a hierarquia de uma rede.

A principal vantagem da utilização desta estrutura em árvore é a facilidade em localizar informações. Depois de se encontrar um objeto, é possível determinar o seu contexto retrocedendo na árvore. Por exemplo, ao pesquisar pelo nome de um funcionário, poderia aceder a informações sobre o mesmo, tais como o seu número de telemóvel, o departamento onde trabalha, etc.

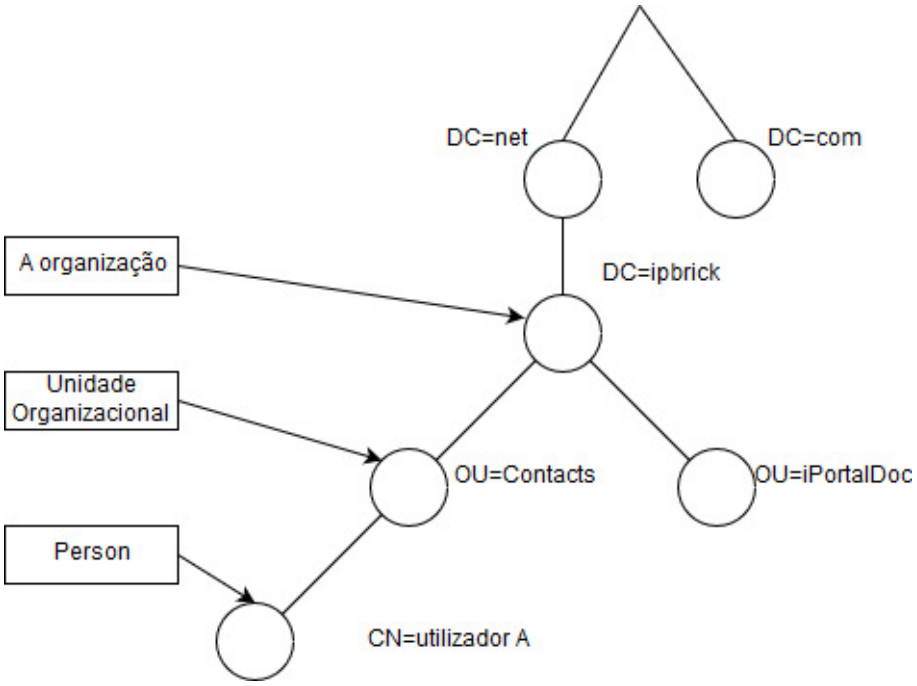

Figura 2.1: Exemplo de árvore de diretório

Cada funcionário pode ter uma conta de acesso no servidor LDAP, para que possa registar dados sobre si e compartilhar arquivos.

O uso de diretórios é frequentemente feito com a finalidade de guardar a informação necessária para identificar ou autenticar os utilizadores, e autorizar o seu acesso a recursos. Na indústria tecnológica de segurança, estes são denominados serviços de segurança de autenticação e autorização.

É importante não confundir o propósito dos servidores e do protocolo LDAP, pois estes não têm como objetivo fornecer serviços de autenticação ou de autorização, mas sim de armazenar e servir os dados. Na figura 2.2 encontra-se o esquema de um pedido de um cliente ao servidor LDAP.

Figura 2.2: Pedido a servidor LDAP

Autenticação e identificação referem-se à forma como é identificado o utilizador que está a tentar aceder a recursos do sistema. Para descobrir quem está a solicitar o acesso aos recursos, é

necessário que o utilizador indique apenas o seu ID, ou indique o seu ID e digite a sua *password*. Depois de se saber quem é que pediu o acesso, é preciso decidir o que é que irá ser permitido ao utilizador fazer. Este é o papel dos serviços de autorização, que podem ser guardados num diretório LDAP [13].

2.3.2 Objetos

Os dados armazenados no *Active Directory* são apresentados ao utilizador de forma hierárquica tal como num sistema de ficheiros. Cada entrada é considerada um objeto. A nível estrutural, existem dois tipos de objetos: *containers* e *non-containers*. Um ou mais *containers* ramificam-se hierarquicamente a partir de um *container* de raiz, e cada *container* pode conter *non-containers* ou outros *containers*. Um *non-container* não pode conter outros objetos.

O tipo de *container* mais utilizado num *Active Directory* é uma unidade organizacional (OU) [11].

2.3.2.1 Identificadores Únicos Globais

Quando, potencialmente, existem milhares de objetos armazenados no *Active Directory*, é fundamental que cada objeto seja unicamente localizável e identificado. Para isso, os objetos têm um identificador global único (GUID), atribuído a cada um, pelo sistema aquando da sua criação.

O GUID trata-se de um número de 128 bits e é a implementação feita pela Microsoft do conceito de identificador universalmente único (UUID), criado pela empresa Digital Equipment Corporation.

Embora um objeto GUID seja resiliente, não é fácil de memorizar e não é baseado na hierarquia de diretórios. Por essa razão, é mais usual referenciar objetos através de nomes distintos (DN) [11].

2.3.2.2 Nomes Distintos

Caminhos hierárquicos em AD são conhecidos como nomes distintos e podem ser referenciados unicamente a um objeto. Nomes distintos são definidos no LDAP como uma forma de referenciar a qualquer objeto num diretório.

Um caminho à raiz da figura 2.1 pode ser "dc=ipbrick,dc=net". Este DN representa o domínio raiz "ipbrick.net".

Um nome distinto relativo (RDN) é o nome utilizado para referenciar unicamente um objeto dentro do seu *container* parente no diretório. Segundo o exemplo da figura 2.1, o DN para a pessoa com o "cn=utilizador A", no *container* "Contacts"no domínio “ipbrick.net” é “cn=utilizador A,cn=Contacts,dc=ipbrick,dc=net”. Então o RDN do utilizador é “cn=utilizador A”.

O RDN é sempre único dentro do seu *container* correspondente. É possível ter dois objetos com o mesmo RDN no diretório, mas apenas se estiverem dentro de *containers* parentes diferentes.

Todos os RDN usam um prefixo que indica a classe do objeto à qual está a ser referenciado. Por defeito, utiliza-se o prefixo “cn”, *common name*.

Tabela 2.1: Tabela com alguns dos tipos de atributos mais utilizados nos servidores de diretório

Prefixo	**Tipo de atributo**
CN	Common Name
O	Organization name
OU	Organizational unit name
DC	Domain component
UID	User ID

Na tabela 2.1 encontram-se alguns dos tipos de atributos mais utilizados nos servidores de diretório [11].

2.3.3 LDIF

Lightweight Directory Import Format (LDIF) é o formato *standard* para diretórios LDAP e é utilizado para exportar e importar dados do diretório.

Os ficheiros LDIF podem ser usados como *backup* e restauro de ficheiros de diretórios.

Uma outra utilização recorrente deste tipo de ficheiros é o preenchimento inicial de diretórios. Isto é, informações de outras fontes podem ser colocadas no formato LDIF e, em seguida, importadas para o diretório LDAP. Geralmente, esse é um processo muito mais rápido do que as alternativas de inserir manualmente ou usar rotinas de inserção em massa.

Por fim, os arquivos LDIF são normalmente usados durante migrações e atualizações para novas versões de software LDAP, ou durante trocas de software de diretórios LDAP, de desenvolvedores diferentes [13].

2.3.4 Listas de Controlo de Acesso

Listas de controlo de acesso (ACL) são listas com permissões ligadas a objetos. Estas permissões definem quem pode ter acesso a um objeto, juntamente com o tipo de acesso no *Active Directory*.

As configurações de permissões são denominadas de entradas de controlo de acesso. Cada entrada pode estar associada a utilizadores, grupos ou processos, e indica a quem é permitido ou negado o acesso a objetos do sistema, que podem ser programas, processos ou ficheiros.

Para além de definir quem tem acesso aos objetos, devem também ser escolhidas as operações que são permitidas efetuar num objeto, tais como apenas leitura, escrita ou pesquisa.

2.4 *Data-centric Security*

Proteger os dados de uma empresa, poderá ser uma tarefa complicada. Os dados podem estar em uso, em movimento ou em repouso.

De acordo com [14], 50 a 60% das empresas não sabem onde se encontram os seus dados e não sabem como os proteger adequadamente. É necessário haver uma segurança baseada nos dados, *Data-centric security*.

Data-centric security é uma abordagem à cibersegurança, que pretende focar-se nos dados sensíveis de uma organização, em vez de se focar na infrastrutura IT. Esta abordagem começa por olhar para os dados do negócio que deverá proteger e porquê, ou por classificar toda a informação existente [15].

Em conformidade com [1], esta abordagem é composta pelas fases enunciadas e detalhadas de seguida.

2.4.1 Descoberta de Dados

Primeiramente, é necessário descobrir onde se encontram os dados.

A descoberta de dados é o processo de análise de ficheiros e pastas, comparando o seu conteúdo com o que a organização definiu como sendo dados sensíveis [16].

Dependendo da indústria, poderão existir vários tipos de dados que, embora não pareçam ser críticos para o funcionamento do negócio, certamente terão um risco associado ao seu compromisso ou perda [1].

Geralmente, os tipos de dados são:

- Dados de recursos humanos de funcionários;
- Dados privados da empresa – planos de negócio, estratégias de aquisição e marcas;
- Dados confidenciais da empresa – localizações e diagramas de rede;
- Dados públicos da empresa – anúncios de produtos e de imprensa;
- Dados de clientes – cartões de crédito e informações pessoais;
- Dados médicos.

Os dados podem estar localizados dentro ou fora da rede de uma empresa. Idealmente, os dados deveriam situar-se apenas dentro da rede da empresa, mas na prática encontram-se em qualquer lugar.

Por exemplo, um funcionário que decida trabalhar em casa, ao enviar os dados que necessita por e-mail para uma conta pessoal, ou fazendo *upload* através dum serviço de armazenamento online, pode fazer com que os dados residam em sistemas e aplicações que não são controladas pela empresa [1].

Geralmente, os dados localizam-se em:

- Redes de partilha;
- Repositório de documentos;

- Sistemas de transferência de ficheiros;
- Parceiros de negócio;
- Computadores dos empregadores e funcionários;
- Telemóveis ou *tablets* pessoais de funcionários;
- Armazenamento portátil pessoal dos empregadores e funcionários;
- Sistemas de armazenamento online;
- Serviços pessoais de e-mail;
- Base de dados;
- Cópias de segurança;
- Hardware reparado/substituído.

2.4.2 Classificação dos Dados

Nesta etapa, em primeiro lugar, deverão ser atribuídos proprietários para cada ficheiro a classificar. Isto é, os proprietários dos dados serão os responsáveis por estes.

De seguida, é preciso classificar os dados.

A classificação é o processo de marcar ficheiros com metadados que indicam quais tipos de informações os ficheiros contêm [16].

Com isto, pretende-se identificar os dados sensíveis de modo a protegê-los adequadamente. É possível classificar de várias formas diferentes, mas seguindo [1], cada ficheiro pode ser caracterizado como sendo "Confidencial restrito", "Confidencial"ou "Público", como exemplificado na figura 2.3.

Os documentos podem ser considerados "Confidencial restrito"quando contenham números de cartões de crédito, números de passaportes, apelidos, endereços, números de segurança social, entre outros. Para os documentos serem considerados "Confidencial", devem conter nomes, datas de nascimento, registos financeiros, etc. Para serem considerados "Público"não poderão conter qualquer tipo de informação sensível.

Uma forma que pode ajudar na classificação dos dados é adotar o pensamento dos atacantes. Isto é, perguntar a nós mesmos o que é que eles podem querer [17].

O nível de proteção de cada ficheiro irá variar conforme a classificação que lhe foi atribuída. No caso de um ficheiro com classificação "Pública", este não terá qualquer tipo de proteção. Um ficheiro com classificação "Confidencial"deverá restringir o seu acesso a um certo grupo de pessoas; poderá, por exemplo, permitir apenas visualizar o documento, proibindo de o editar. A classificação "Confidencial restrito"implicará para o documento ter um nível de segurança mais sofisticado, podendo o seu acesso se concretizar apenas com uma *password*.

Após a criação do modelo de classificação, este deverá ser adotado por toda a empresa.

	Restricted confidential (Level 1)	Confidential (Level 2)	Public (Level 3)
Data type	Customer: • CC# • PII Employee: • SSN# • PII Company: • Merger Plans • New product	Customer: • PII Employee: • PII Company: • Internal documents	• Anything not in the previous sections. • Items considered to be available in the public domain.
Data protection	Data encryption, hashing, or tokenization	Restricted access permissions	None

Figura 2.3: Modelo de classificação de dados [1]

2.4.3 Gerir Acesso aos Dados

Nesta fase, é definido quem terá acesso aos ficheiros. Para isso, seguiremos uma abordagem chamada *role-based access control* (RBAC).

Role-based access control é uma abordagem de segurança que permite restringir o acesso a um sistema, com base na função da pessoa na organização [18]. Ou seja, os funcionários deverão poder aceder apenas à informação que necessitam para realizarem o seu trabalho.

Uma ferramenta RBAC poderá:

- Limitar os documentos que um certo conjunto de funcionários (com as mesmas funções na organização) pode gerir;
- Adicionar e remover membros;
- Atribuir funções a um grupo.

Ao adicionar um utilizador a um ou mais grupos, este deverá ter acesso a todas as funções desse(s) grupo(s). Assim que deixar de precisar de ter acesso a essas funções, deverá ser excluído do grupo, restringindo o seu acesso.

Ao implementar esta abordagem numa organização, é expectável que reduza o trabalho administrativo e o suporte IT, pois deixa de ser necessário alterar *passwords* e escrever documentos a cada vez que um utilizador muda de função na organização.

2.4.4 Métodos de Prevenção de Perda de Dados

Data Loss Prevention (DLP) é uma ferramenta que pretende reforçar a proteção de dados classificados anteriormente pela empresa [1]. DLP pode ajudar a encontrar dados em várias localizações, reforçar a encriptação, bloquear transmissões inseguras, cópias e armazenamento de

dados não autorizados, com base na sua classificação. As soluções DLP ajudam as organizações a reportar transgressões no seu sistema, pois é capaz de determinar o destino dos dados e a extensão do ataque [19].

2.4.4.1 Dados em Repouso

Os dados podem ser armazenados em diversas localizações dentro de uma rede de uma empresa, tais como em partilhas na rede, bases de dados, repositórios de documentos, armazenamento *online* e dispositivos de armazenamento portátil.

De acordo com [1], a maioria das soluções DLP é capaz de examinar armazenamentos de dados e de fornecer um agente que possa ser implementado em sistemas finais, para monitorizar e impedir ações não autorizadas em dados confidenciais.

Para proteger os dados confidenciais descobertos, deverá ser utilizada uma função automática (da solução DLP), que é capaz de mover os dados para um local seguro.

2.4.4.2 Dados em Uso

Dados em uso são dados que são ativamente processados numa aplicação, processo, memória ou outro local temporariamente, durante a duração duma função ou transição [1]. Aplicações *web* capazes de ler, adicionar, remover e modificar dados, são um exemplo de dados em uso.

Os dados em uso podem ser monitorizados por um agente instalado no sistema final que permite apenas certos tipos de uso dos dados, proibindo guardar os ficheiros localmente ou enviá-los via e-mail. O agente DLP, instalado no sistema final, deverá ser inserido em baixo da pilha TCP/IP para garantir que ele possa detetar os dados antes de qualquer encriptação poder ser aplicada.

Ao utilizar uma solução DLP *Endpoint* é possível limitar os locais que os dados podem ser armazenados, como é que podem ser transmitidos, e quais as aplicações que podem interagir com estes.

2.4.4.3 Dados em Movimento

Dados em movimento são dados que estão a ser movidos dum sistema para outro, localmente ou remotamente, como sistemas de transferência de ficheiros, e-mail e aplicações *web* [1].

Na maioria das organizações existem vários métodos de comunicação disponíveis, como e-mail, transferência de ficheiros, mensagens instantâneas e serviços de conferência por voz, vídeo ou mensagens instantâneas. Foram desenvolvidas soluções DLP capazes de intercetar e desencriptar comunicações, para descobrir dados classificados. Existem soluções para HTTP/HTTPS, FTP, SMTP, interseções IM, e inspeções.

Na figura 2.4 é dado um exemplo duma solução *Network* DLP, implementada para e-mail, web, e interseção, inspeção e mitigação de tráfego geral numa rede.

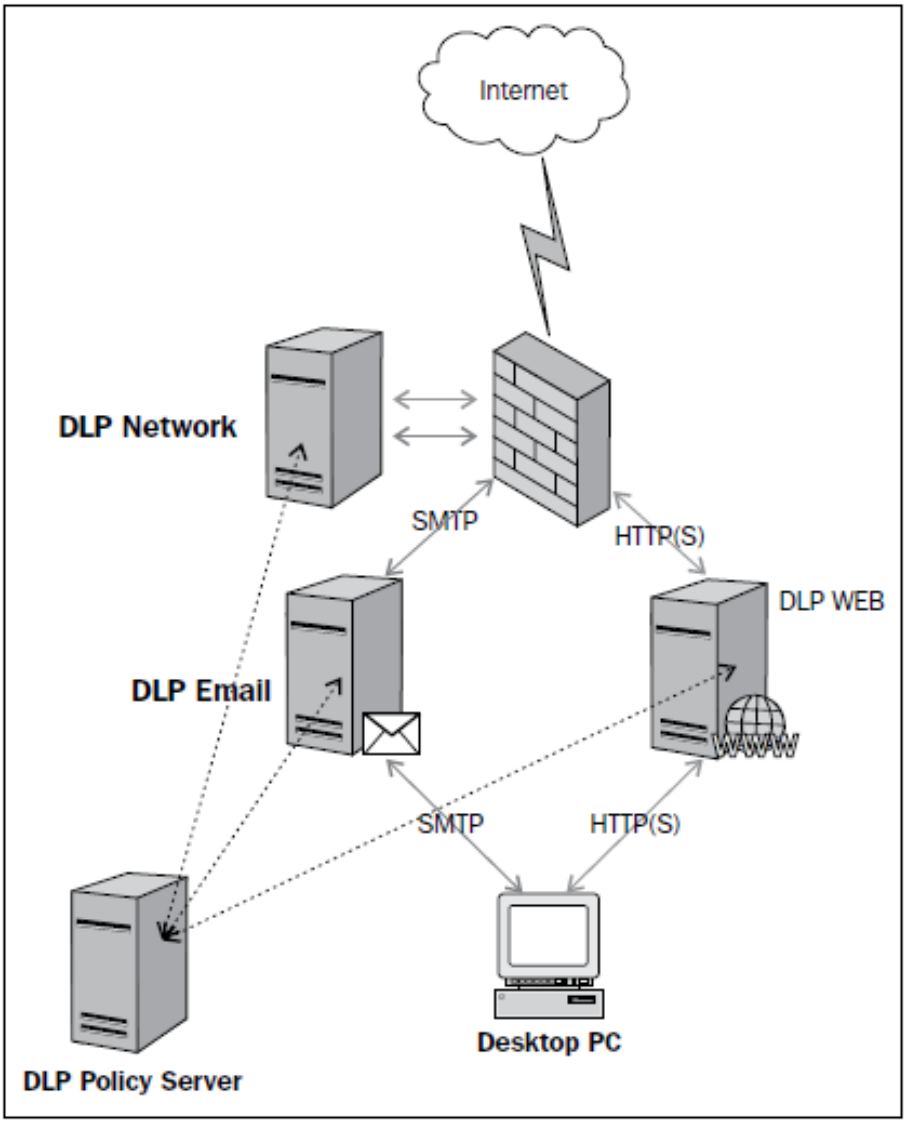

Figura 2.4: Exemplo de solução *Network* DLP [1]

Quando o tráfego está a sair da rede, através de qualquer método de comunicação mencionado anteriormente, é enviado para o DLP, desencriptado (se necessário), e inspecionado. O DLP poderá bloquear, permitir ou encriptar os dados. O DLP pode ser ativado pelo tipo de dados, pelo conjunto fonte/destino, por remetentes e pelos recetores.

Como, geralmente, existe menos confiança quando são envolvidas entidades e redes externas do que violações internas, é expectável que os dados em transição sejam a prioridade das organizações, no que diz respeito à prevenção de perda de dados.

Ao implementar *Network* DLP, E-mail DLP, *Web* DLP e *Endpoint* DLP, a maioria dos cenários de perda de dados será detetado e poderá ser prevenido, com um alto nível de sucesso. No entanto, cada organização deverá entender qual o nível de proteção mais adequado de acordo com o que necessitam.

2.4.4.4 *Network* DLP

Network DLP é uma tecnologia que protege as comunicações na rede de uma organização, incluindo e-mail e aplicações *web* [20]. Isto é, *Network* DLP pode conter E-mail DLP e *Web* DLP.

Soluções *Network* DLP têm o objetivo de evitar que informações sensíveis saiam da organização, através da rede. Podem ser capazes de:

- Inspecionar e controlar o tráfego de e-mail, *webmail*, aplicações *web*, HTTP/S, FTP/S e TCP/IP;
- Prevenir a perda de dados sensíveis através da rede, independentemente da porta ou protocolo;
- Inspecionar assuntos, mensagens e anexos em e-mails;
- Aplicar a monitorização e o bloqueio (com base em políticas) de aplicações *web*;
- Encriptar conteúdo de e-mail para comunicações seguras e cumprimento regulamentar;
- Notificar utilizadores e administradores quando o tráfego da rede violar as políticas corporativas de proteção de dados.

O ponto negativo destas soluções é que apenas funcionam dentro da rede da organização. Fora da rede, não é possível ter visibilidade sobre o que está a acontecer aos dados.

2.4.4.5 *Endpoint* DLP

Com *Endpoint* DLP, um agente é instalado no fim de cada sistema, e fornece visibilidade nos dados à medida que são criados. Por exemplo, quando um documento com números da segurança social é criado, este ficheiro pode alertar que contém dados sensíveis. O agente também consegue detetar processos como "*copy* + *paste*"e imprimir, impedindo que os dados possam ser transmitidos para dispositivos USB.

Neste caso, pelo facto do agente estar instalado num *endpoint*, estará sempre a proteger os dados, ainda que o computador não esteja dentro da rede da organização.

Com esta solução, cada computador precisa que o agente seja instalado e no futuro atualizado. Caso a haja um número elevado de computadores e servidores ou estes estejam espalhados demograficamente dentro da rede de uma organização, implementar esta solução pode ser um tarefa complexa.

Para escolher entre soluções *Network* DLP ou *Endpoint* DLP, devemos ter em conta o nível de controlo do *endpoint* que a organização tem: se tem pouco controlo, uma solução *Network* DLP é o mais adequado, caso contrário, deve-se aplicar uma solução *Endpoint* DLP.

Segundo [16], DLP é mais eficaz quando integrado com uma solução de encriptação ou gestão de chaves.

2.4.5 Métodos de Proteção de Dados

2.4.5.1 Encriptação

Encriptação é o método de transformar informação, usando um algoritmo, de modo a impossibilitar a sua leitura a todos, exceto aqueles que possuam uma identificação particular (geralmente uma chave). O resultado deste processo é uma informação encriptada.

Existem dois tipos de encriptação: simétrico e assimétrico. Ambos requerem uma chave para encriptação e desencriptação. Essencialmente, na encriptação simétrica, as chaves de encriptação e desencriptação são as mesmas. No caso da encriptação assimétrica, utilizam-se duas chaves distintas: a chave de encriptação e a de desencriptação. A chave de encriptação é publicada para qualquer um usar para encriptar suas mensagens. Porém, somente o grupo destinatário tem acesso à chave de desencriptação, que é secreta e que permite que as mensagens sejam lidas.

O que torna uma encriptação mais segura que outra é o tamanho da chave, rondas de encriptação, e *salting* (falsificação). Rondas de encriptação refere-se ao número de vezes que os dados são executados no algoritmo de encriptação até gerar o texto cifrado. *Salting* é o processo de adicionar *bits* aleatoriamente ou uma *password* no processo de encriptação, para aumentar o nível de segurança do texto cifrado.

Existem vários algoritmos de encriptação, sendo o AES-256 um dos recomendados.

2.4.5.2 GPG

Outro tipo de encriptação é *Gnu Privacy Guard* (GPG). Esta é uma implementação *open source* da solução comercial de encriptação de chave assimétrica PGP.

Imaginemos que se pretende enviar um e-mail da pessoa A para a pessoa B. Para a mensagem ser enviada de forma segura, a pessoa B terá de fornecer a sua chave pública à pessoa A. Esta chave pública será utilizada para encriptar a mensagem, pela pessoa A. Depois de encriptada com a chave pública da pessoa B, esta mensagem só pode ser desencriptada utilizando a chave privada da pessoa B. Então, a pessoa A envia a mensagem para a pessoa B, sendo que esta insere a sua chave privada, desencriptando a mensagem.

Este método impede que a mensagem possa ser lida por outra pessoa para além das duas envolvidas, pois, mesmo que intercetem a mensagem encriptada ou a até mesmo a chave pública, nunca a irão decifrar por não terem a chave privada, que é a única que permite decifrar a mensagem.

Na figura 2.5 é possível entender todos os passos deste método.

2.4.5.3 *Tokenization*

Em *data-centric security* um *token* é fornecido em vez de dados sensíveis. Este método é muito utilizado em sistemas de processamento de cartões de crédito, para substituir os números do próprio cartão. Ao substituir os dados sensíveis por um *token*, os dados originais são guardados numa base de dados segura, separada dos sistemas de negócio [21]. *Detokenization* é o

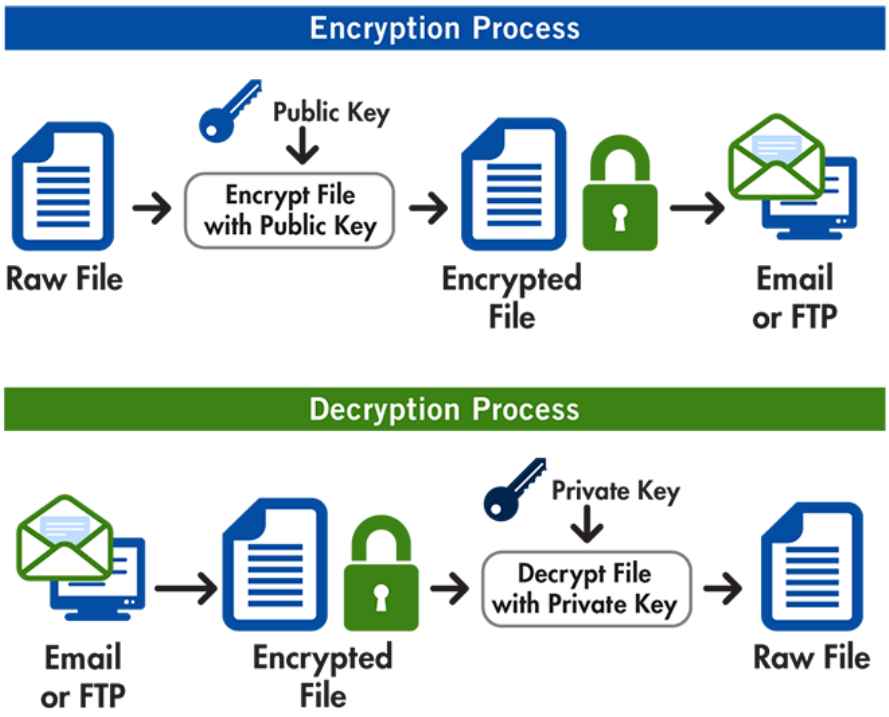

Figura 2.5: Como funciona GPG/PGP [2]

processo reversivo e só pode ser executado pela plataforma original de *tokenization*. Desta forma, os sistemas nunca armazenam, processam ou transmitem o *pin*, apenas o *token* [22].

Um *token* é um número aleatório, sem relação matemática com o número original e sem valor para além da referência com o original, numa base de dados mais segura. Como *tokenization* não é encriptação, não existe nenhum método reversivo que permita recuperar os dados fora dum sistema *tokenization*.

Atualmente, os *tokens* são utilizados também para ficheiros JSON ou XML e em páginas *web*. Algumas soluções de *tokenization* substituem dados armazenados em bases de dados, enquanto outras trabalham em fluxos de dados. Isto permite que o método funcione para dados simples e complexos, em repouso ou em movimento.

2.4.5.4 *Data Masking*

Segundo [1], *Data Masking* deve ser utilizado apenas para restrições de visualização em sistemas e em registo de *logs*, não devendo ser considerado realmente um método de proteção de dados.

Esta ferramenta permite, de uma forma segura, esconder informação sensível, substituindo essa informação por outra aleatória, mas ao mesmo tempo realística – ver figura 2.6.

Por exemplo [21], podemos substituir o número da segurança social de um indivíduo por outro número aleatório, ou um nome aleatório selecionado a partir de uma lista telefónica, mas manter

género. Podemos trocar a data de nascimento por um valor aleatório dentro de x dias em relação ao valor original, preservando efetivamente a idade. Desta forma, a informação original sensível é removida sem alterar completamente o valor dos dados, possibilitando futuras análises.

Figura 2.6: Exemplo de *Data Masking* [3]

Existem diversas abordagens para a implementação deste método, sendo que *Dynamic Data Masking* (DDM) é uma das mais recomendadas para proteger dados em aplicações *web*.

Dynamic Data Masking é o processo de mascarar, baralhar, esconder, auditar, ou bloquear o acesso a dados ao nível do utilizador. Esta solução é um *software proxy* situado num único servidor na junção de aplicações de negócios, ferramentas de relatório e desenvolvimento, e bases de dados [4].

Quando um utilizador solicita dados, as suas credenciais e informações de sessão são examinadas pela plataforma de *masking*. Utilizadores autorizados terão acesso aos dados originais. Utilizadores não autorizados a aceder a informação sensível, ou não aprovados a usar a aplicação, receberão dados mascarados. Tudo isto é realizado dinamicamente, em tempo real.

Na figura 2.7 é apresentado um exemplo com três tipos de utilizadores a tentarem aceder a dados. O utilizador à esquerda é autorizador a ver todo o tipo de informações. No centro, está um funcionário com autorização apenas para ver os dados mascarados, para que possa executar as suas tarefas. Por fim, à direita, está um utilizador que necessita de informação no formato apropriado, para cumprir o seu trabalho, e receberá valores baralhados para cumprir com a regulação de proteção de privacidade.

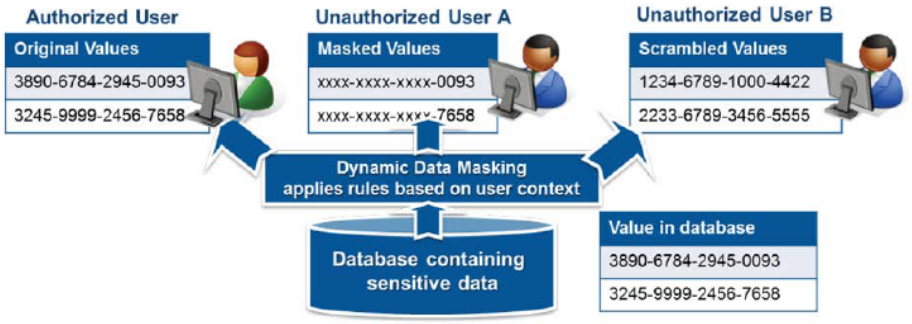

Figura 2.7: Exemplo de interações num sistema com *Dynamic Data Masking* [4]

2.5 Exemplos de Aplicações

Serão agora apresentados exemplos de aplicações que utilizam algumas das ferramentas apresentadas.

2.5.1 Symantec

A Symantec apresenta diversas soluções de segurança, baseadas em *data-centric security*, ao que chamam de *Information Centric Security*.

Com *Information Centric Security*, é possível proteger dados sensíveis, onde quer que estejam, assegurando o acesso a apenas utilizadores autorizados. É possível prevenir o acesso a utilizadores através de controlo e monitorização central.

A solução *Data Loss Prevention* da Symantec fornece um alto nível de proteção, que previne violação de dados. Com tecnologia líder da indústria, é possível descobrir, monitorizar e utilizar recursos de proteção que dão total visibilidade e controlo sobre os dados confidenciais [23].

É aplicado *Endpoint DLP*, para a descoberta e para a prevenção.

Em relação à rede, o *Network DLP* é utilizado para capturar e analisar tráfego dentro da rede da empresa, ou seja monitoriza. Para além disso, *Network DLP* é utilizado para prevenção em e-mail e na *web*.

No que diz respeito ao armazenamento, o *Symantec DLP Network Discover* encontra dados confidenciais ao examinar partilhas de ficheiros na rede, bases de dados, e outros repositórios da organização. O *Symantec DLP Network Protect* protege automaticamente os ficheiros expostos detetados e oferece opções, tais como mover os ficheiros, colocá-los em quarentena, ou encriptá-los.

No que se refere à *cloud*, o *Symantec DLP Cloud Detection Service* inspeciona conteúdo extraído de aplicações *cloud* e tráfego *web*. *Symantec DLP Cloud Service for Email* monitoriza, em tempo real, tráfego do e-mail corporativo.

2.5.2 Thales

Thales dispõe de uma solução que integra *tokenization* e *dynamic data masking* numa só. Assim, uma organização será capaz de proteger com eficiência e tornar anónimos os ativos sensíveis e registos de portadores de cartões, estejam eles em ambientes *data center*, *big data* ou na *cloud*.

Esta solução reduz drasticamente o custo e o esforço requeridos para estar em conformidade com políticas de segurança e regulamentos como PCI-DSS [24].

São vários os benefícios desta abordagem, tais como:

- Promover inovação sem introduzir risco – *tokenization* dos dados e manter o controlo e a conformidade ao migrar para ambientes *cloud* e *big data*;
- Escala global – Implementar a solução globalmente sem preocupações com sincronização de *token*, desempenho ou custos não controlados.

2.6 Sistemas de Gestão Documental e de Processos

Atualmente, para gerir eficazmente uma empresa que trabalhe com uma quantidade avultada de documentos, é crucial adotar sistemas que consigam gerir corretamente todas as informações e documentos que fazem parte da mesma.

Um sistema de gestão documental (DMS) é, portanto, um sistema que é capaz de guardar, controlar, coordenar, processar e/ou reaver documentos que estejam num formato eletrónico ou a partir de imagens digitalizadas de documentos em papel. Para além disso, um DMS digital pode ser definido como uma aplicação de software que recolhe documentos para o armazenamento, recuperação e arquivamento seguros dos mesmos [25].

As principais características que um sistema deste tipo deverá apresentar são [26]:

- Entrada de documentos – A maioria dos negócios utiliza documentos em formato de papel e digital. Idealmente, um DMS deverá permitir a entrada de documentos através de e-mail, *scanner*, *upload* manual, *upload* em massa, aplicações moveis e serviços *web*;
- Indexação de documentos – É o processo de marcar ou associar documentos a diferentes termos de pesquisa. A indexação é um caminho para os documentos. Esse caminho é baseado nos processos do negócio. Todos os DMS têm algum tipo de sistema de indexação, sendo que os mais básicos identificam os documentos pela data ou tipo de ficheiro;
- Pesquisa de documentos – Não importa os índices que são utilizados, o poder da indexação de documentos é revelado quando fazemos uma pesquisa. Um motor de pesquisa documental deverá garantir uma pesquisa segura e eficaz, permitir uma pesquisa avançada em todos os atributos de documentos, e deverá ser um motor de pesquisa documental escalável;
- Processamento de documentos – Envolve a conversão de texto digitado e manuscrito de documentos eletrónicos e de papel, em informação eletrónica utilizando reconhecimento inteligente de caracteres ou reconhecimento ótico de caracteres. Idealmente, um DMS deverá permitir criar documentos usando *templates*, guardar as várias versões modificadas do mesmo ficheiro, reencaminhar, mover e partilhar documentos, e ter um editor de documentos embutido para vários tipos de ficheiros;
- Automação de *workflow* – Um bom sistema de gestão de documentos deve ter incorporado um gestor de processos de negócio e um Workflow automático de nível corporativo que encaminha automaticamente os documentos para seu destino. A automação de *workflow* deve fornecer um processamento baseado em regras para documentos recebidos, configurar regras individuais e ações do documento, providenciar um *workflow* manual e automático e criar automaticamente registos com base em documentos;
- Segurança de documentos – A segurança é um dos aspetos mais críticos de um DMS. O software ideal fornecerá um alto nível de criptografia de documentos e acesso baseado em funções na organização, bem como direitos de acesso avançados, documentos encriptados no sistema de ficheiros e indexação de todas as revisões;

- Interface *user-friendly*– Uma interface de um DMS deverá ser simples e de fácil navegação. Alertas e notificações, calendário e e-mail embutido, e caixas de entrada de documentos, são alguns exemplos de ferramentas que uma interface de um DMS deverá conter;

- Personalização – Cada negócio tem uma necessidade diferente para utilizar um DMS, pois cada um tem requisitos diferentes. Por esta razão, um DMS deverá ter um certo nível de personalização, permitindo aos utilizadores criar janelas e registos costumizados, adicionar atributos personalizados a documentos e, entre outros, criar registos *dashboard* personalizados.

2.7 Tecnologias

2.7.1 HTML

HTML é uma linguagem de marcação, utilizada para a criação de páginas *web*.

Para criar e editar ficheiros nesta linguagem, basta um simples editor de texto, sendo a extensão do ficheiro *.html*. Para visualizar o documento, pode ser utilizado qualquer *browser*.

Na figura 2.8 encontra-se do lado esquerdo um código HTML exemplo, e no lado direito, a sua representação num *browser*.

```
<!DOCTYPE html>
<html>
<body>

<p>This is a paragraph!</p>

<h2>An Unordered HTML List</h2>

<ul>
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ul>

<h2>An Ordered HTML List</h2>

<ol>
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ol>

</body>
</html>
```

This is a paragraph!

An Unordered HTML List

- Coffee
- Tea
- Milk

An Ordered HTML List

1. Coffee
2. Tea
3. Milk

Figura 2.8: Código HTML exemplo e respetiva representação

2.7.2 PHP

O PHP é uma linguagem de *script open source* de uso geral, especialmente adequada para o desenvolvimento *web* e que pode ser embutida dentro do HTML.

O código é executado no servidor, gerando o HTML que é então enviado para o *browser*. O *browser* recebe os resultados da execução desse script, mas não sabe qual era o código fonte.

Na figura 2.9 encontra-se do lado esquerdo um código PHP exemplo, e no lado direito, a sua representação num *browser*.

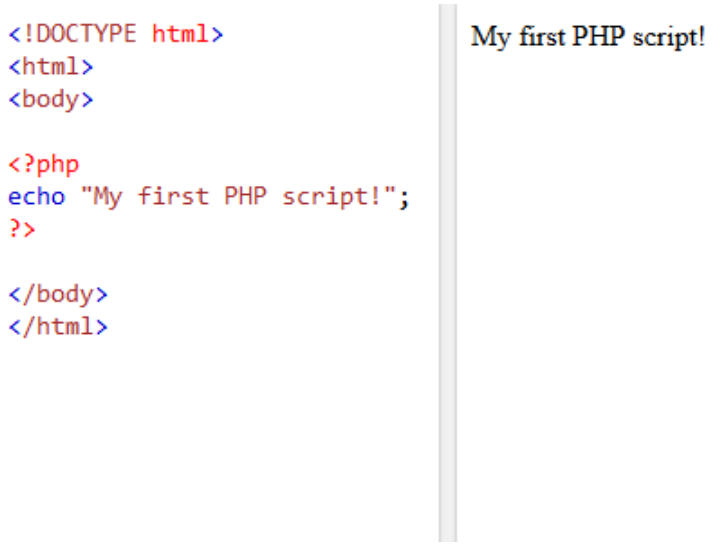

Figura 2.9: Código PHP exemplo e respetiva representação

De acordo com [27], PHP é capaz de:

- Gerar conteúdo de página dinâmico;
- Criar, abrir, ler, escrever, apagar e fechar arquivos no servidor;
- Enviar e receber *cookies*;
- Adicionar, apagar, modificar dados numa base de dados;
- Controlar o acesso de a utilizadores;
- Encriptar dados.

2.7.3 JavaScript

JavaScript é uma linguagem de programação orientada a objetos, usada para tornar páginas da *web* interativas.

Dentro de um ambiente *host* (por exemplo, um navegador da Web), o JavaScript pode ser conectado aos objetos do seu ambiente para fornecer controlo programático sobre eles.

Através de JavaScript, é possível que as *scripts* possam ser executadas do lado do cliente. Isto é, será possível criar uma interação com a página *web* sem que o utilizador tenha de atualizar a página para obter atualizações.

Na figura 2.10 encontra-se do lado esquerdo um código JavaScript exemplo, e no lado direito, a sua representação num *browser*.

```
<!DOCTYPE html>
<html>
<body>

<h2>JavaScript Functions</h2>

<p>This example calls a function
which performs a calculation, and
returns the result:</p>

<p id="demo"></p>

<script>
function myFunction(p1, p2) {
    return p1 * p2;
}
document.getElementById("demo").inner
HTML = myFunction(4, 3);
</script>

</body>
</html>
```

JavaScript Functions

This example calls a function which performs a calculation, and returns the result:

12

Figura 2.10: Código JavaScript exemplo e respetiva representação

2.7.4 PostgreSQL

PostgreSQL é um sistema *open source* de base de dados, do tipo objeto-relacional, que utiliza e estende a linguagem SQL.

PostgreSQL suporta diversos recursos, para além do *standard* SQL, tais como:

- *Query's* complexas;
- Chaves estrangeiras;
- Integridade transacional.

2.8 *Software*

A IPBRICK SA é uma empresa fabricante e distribuidora de software para Comunicações Empresariais. O seu principal foco é em Comunicações Unificadas, Correio Eletrónico e Ferramentas Colaborativas, Gestão de Documentos e Processos, e numa Rede Social Corporativa [28].

Nesta dissertação, para além do Sistema Operativo IPBRICK (IPBRICK OS), também irá ser necessário utilizar duas aplicações da empresa: o iPortalDoc e o Contacts.

2.8.1 IPBRICK OS

O IPBRICK OS é um sistema operativo baseado em Linux Debian, desenvolvido pela empresa IPBRICK SA. Esta plataforma de comunicações para empresas dá suporte às soluções de:

- Comunicações Unificadas sobre IP;
- Gestão de Documentos e de Processos;
- Email e Ferramentas Colaborativas;
- Rede Social Corporativa.

Ao instalar o IPBRICK OS, será disponibilizado um servidor que dispõe de três grupos de funcionalidades [29]:

- **Intranet** - *Mail & Groupware*, Rede Social Corporativa, Servidor de Ficheiros, Servidor de Impressão, Controlador de Domínio, LDAP, DNS e DHCP;
- **Comunicações Unificadas sobre IP** - voz, vídeo, mensagens instantâneas, fax e SMS;
- **Segurança** - Firewall, VPN, Proxy, Mail Relay, Filtragem de Conteúdo e Servidor Web.

2.8.2 iPortalDoc

O iPortalDoc é um Sistema de Gestão Documental e de Processos, baseado em *workflows*, que funciona como um serviço de valor acrescentado para Intranet. Esta aplicação permite o registo, classificação e tratamento de todas as comunicações, bem como o tratamento processual de todos os processos de uma organização.

A qualquer momento de um determinado processo, que decorra no iPortalDoc, haverá sempre acesso a todo o histórico de pessoas implicadas, intervenções realizadas, bem como documentos e emails associados, facilitando a pesquisa e evitando perdas de tempo e de informação [30].

Este *software* tem várias vantagens associadas, tais como:

- Automatização e uniformização dos processos de trabalho;
- Desmaterialização da documentação e dos processos;
- Aumento da eficiência administrativa e processual;
- Gestão do arquivo da empresa de forma centralizada.

2.8.3 Contacts

O Contacts é utilizado através do iPortalDoc para a gestão de entidades e contactos. Tem uma interface desenvolvida a pensar na gestão dos contactos das empresas e integra com o e-mail de cliente e com o iPortalDoc [31].

Esta aplicação permite aceder a múltiplas informações de cada contacto, tais como a morada, o contacto telefónico e o e-mail. Para além dessa informação, cada contacto é classificado pelo seu tipo de entidade; pode ser, por exemplo, um contacto internacional e/ou parceiro.

2.9 Resumo

No capítulo 2, foi explicado o conceito de *Active Directory*, LDAP e *Data-centric security*, apresentando as fases que constituem este tipo de segurança. Foram apresentados exemplos de aplicações existentes no mercado, que têm como base este tipo de segurança. Foram também fornecidas bases teóricas sobre as tecnologias que irão ser utilizadas e foi feita uma introdução ás aplicações da IPBRICK SA.

Capítulo 3

Caracterização do Problema

No presente capítulo, começaremos pela definição geral do problema. Depois, serão detalhados os problemas para cada fase, de cada aplicação, que serão necessários resolver.

3.1 Definição do Problema

A IPBRICK SA desenvolve soluções para as comunicações empresariais, focando-se em comunicações unificadas, ferramentas colaborativas e na gestão de documentos e processos. Por trabalhar diariamente com diversos clientes de diferentes ramos, é frequente o pedido de melhoria nas aplicações desenvolvidas pela empresa. Sendo que a empresa tem um espírito de melhoria contínua, não são apenas os clientes que detetam problemas a corrigir e melhorias a implementar, mas também os próprios desenvolvedores de *software*.

Esta dissertação irá focar-se nas melhorias a desenvolver e problemas a corrigir, relacionados com a confidencialidade e a segurança da informação. Estes dois tópicos são de grande importância para uma empresa que desenvolve soluções de comunicações, pois são características chave que podem fazer a diferença entre convencer ou não um cliente a querer utilizar este tipo de soluções na sua organização.

O problema desta dissertação foi dividido em duas partes: a confidencialidade e a segurança da informação. A confidencialidade será o principal tema a considerar no Contacts, a aplicação de gestão de entidades e contactos da IPBRICK SA. No iPortalDoc, a aplicação de gestão documental e de processos da IPBRICK SA, irá ser dada maior atenção à segurança da informação que esta contém.

3.1.1 Contacts

O objetivo principal da primeira parte da dissertação é implementar duas melhorias no Contacts e corrigir um problema relacionado com a confidencialidade de entidades e contactos da IPBRICK. Nesse sentido, a primeira parte da dissertação foi sub-dividida em três partes.

Esta aplicação tem como objetivo principal gerir e apresentar dados de entidades e contactos aos funcionários de uma organização. Nem todos os funcionários necessitam de ter acesso a todos

os dados presentes na aplicação e, por essa razão, é necessário gerir as permissões de acesso a esses dados.

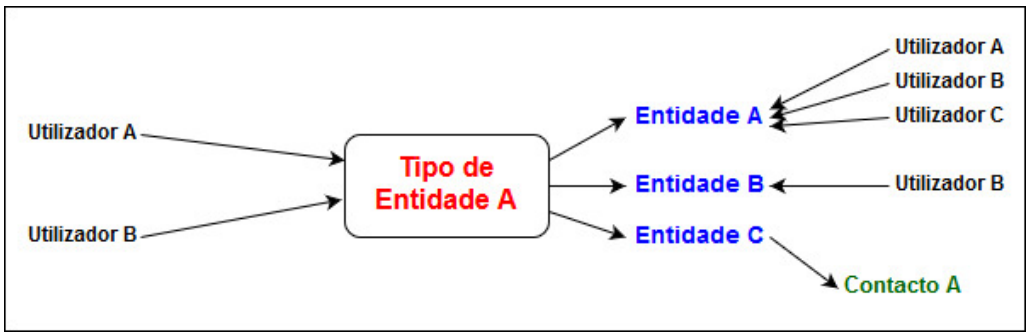

Figura 3.1: Diagrama da organização do Contacts

O diagrama da figura 3.1 foi desenvolvido para clarificar a organização do Contacts, antes de avançarmos para o problema. Este diagrama demonstra que na aplicação existem tipos de entidades, entidades, contactos e utilizadores. Cada entidade tem, obrigatoriamente, pelo menos um tipo de entidade associado (por exemplo, fornecedor, parceiro, funcionário, etc). Cada entidade pode ter contactos associados. Os utilizadores têm sempre um perfil de utilizador associado, que pode ser personalizado, permitindo definir, por exemplo, se o utilizador tem permissões de visualização, edição ou criação. Cada utilizador pode estar associado a entidades ou a tipos de entidade, sendo que no caso de estar associado a um tipo de entidade, irá ter permissões de acesso a todas as entidades e contactos desse tipo, conforme o seu perfil.

Em seguida serão apresentadas as três partes do problema da dissertação, relativo ao Contacts.

3.1.1.1 Permissões de Acesso a Entidades

Numa primeira fase, foi pedido que fosse implementada uma melhoria relativa às permissões dos utilizadores do Contacts. Este não foi um pedido por parte de clientes da IPBRICK com poucas entidades e poucos contactos, mas sim por aqueles que têm dezenas ou centenas de contactos na sua rede empresarial.

Como já foi explicado, um utilizador do Contacts pode estar associado a um tipo de entidade ou diretamente associado a entidades. Se um tipo de entidade não tiver nenhum utilizador com permissões associadas a esse tipo e, simultaneamente, se esse tipo de entidade não tiver nenhum utilizador associado a nenhuma entidade pertencente a esse tipo, então este tipo de entidade deixará de estar sujeito a restrição de permissões. Ou seja, o objetivo é que todos os utilizadores do Contacts, tendo em conta o seu perfil, passem a ter acesso automático às entidades e contactos associados a tipos de entidade que obedeçam às condições referidas.

Tendo sempre em mente a confidencialidade das entidades e contactos, esta melhoria tem o principal intuito de reduzir o tempo de atribuição de permissões aos utilizadores. A resolução desta tarefa irá permitir a empresas como a própria IPBRICK SA, que tem milhares de entidades

e contactos na sua rede, automatizar uma parte do processo de atribuição de permissões de utilizadores, a entidades e contactos. Sempre que forem cumpridas as condições, este processo passará a ser feito de forma automática, e como consequência, os erros de atribuição de permissões a utilizadores irão diminuir, aumentando a proteção relativa à confidencialidade dos dados das entidades e contactos.

3.1.1.2 Confidencialidade das Entidades

Novamente de encontro às permissões de acesso a entidades, por parte dos utilizadores, surgiu outra melhoria necessária de ser implementada no Contacts. Até ao momento, existem três formas diferentes de atribuição de permissões aos utilizadores da aplicação. É possível associar utilizadores a tipos de entidade, selecionando o utilizador e depois o tipo de entidade; outra forma é associar entidades a utilizadores, selecionando a entidade e depois o utilizador; e ainda, é possível associar utilizadores a entidades, escolhendo-se o utilizador e só depois a entidade.

O problema é que nestas três formas de atribuição de permissões só existe um método para filtrar as entidades que pretendemos. Este método limita-se a filtrar as entidades por tipo de entidade ou por país. Porém, também existe uma barra de pesquisa que permite pesquisar entidades e utilizadores, por nome. Ora, tendo em conta estas ferramentas existentes na aplicação, atribuir permissões a um conjunto de utilizadores, em listas de centenas de utilizadores, pode-se tornar algo quase inviável para uma organização de grande dimensão, uma vez que é preciso pesquisar e selecionar os utilizadores, um a um.

Assim, o objetivo é facilitar o processo de atribuição de permissões em bloco. Para isso, deverá ser desenvolvida e implementada uma nova ferramenta que permita pesquisar por todos os utilizadores associados a um tipo de entidade ou a uma entidade específica.

Embora a implementação desta melhoria tenha mais importância para empresas que lidam com muitas entidades e contactos, esta tarefa não só visa reduzir os erros de atribuição de permissões (tendo um impacto positivo no que diz respeito à confidencialidade), como também reduzir o tempo que é necessário para encontrar os utilizadores a quem queremos atribuir permissões.

3.1.1.3 Conformidade com a Confidencialidade de Entidades e Contactos

A terceira fase do projeto incide sobre um problema que envolve a forma como as permissões do Contacts são transmitidas para outras aplicações que acedem ao *groupware* da IPBRICK.

O IPBRICK.MAIL é uma aplicação da IPBRICK SA para e-mail e ferramentas colaborativas. Esta aplicação possibilita gerir e-mail, contactos, agenda e tarefas. Facilmente integramos o nosso email de trabalho nesta aplicação, ficando com a lista de contactos de utilizadores e de entidades disponíveis para acedermos. Embora exista o IPBRICK.MAIL que proporciona todas estas funções referidas, é possível utilizar outras aplicações do mercado, tais como o Thunderbird e o Outlook, para aceder a toda a informação.

O lógico seria que as permissões que foram definidas no Contacts se mantivessem e fossem transmitidas para as aplicações de e-mail, que contêm contactos da empresa. O que é facto é que

isso não acontece, surgindo um sério problema: atualmente quando a sincronização dos dados para o LDAP está ativa, o Contacts passa a informação toda, sendo que a mesma depois fica disponível para todos os utilizadores que acederem à mesma via *Groupware*.

Com este problema, a confidencialidade das informações das entidades é posta em causa, pois se existe uma limitação de acesso aos utilizadores no Contacts, também tem obrigatoriamente de existir a mesma limitação em aplicações que requisitem os contactos e entidades do *groupware*, neste caso da IPBRICK.

O objetivo desta fase é que o LDAP, que até agora apenas servia para gerir as passwords de cada utilizador (representado na figura 3.2), passe a gerir também as entidades e tipos de entidade que cada utilizador tem acesso, segundo o que está definido no Contacts.

Figura 3.2: Exemplo de um pedido LDAP

3.1.2 iPortalDoc

A IPBRICK SA pretende que seja elaborada uma solução que permita proteger os ficheiros mais importantes que se encontrem no iPortalDoc, de forma a que só possam ser acedidos por utilizadores a quem lhes foi dado permissões para tal. Assim, o principal objetivo da segunda parte da dissertação é elaborar uma solução que permita proteger documentos que estão no armazém do iPortalDoc.

Para cumprir este objetivo, há que ter em conta um requisito imposto pela IPBRICK: todo o *software* que faça parte da solução, terá que ser gratuito e *open-source*.

Atualmente, o iPortalDoc já permite restringir quem pode aceder aos ficheiros, através da escolha de permissões de utilizadores. Por si só, esta funcionalidade já eleva a segurança da aplicação. Porém, quando existem ficheiros que apenas devem ser utilizados na rede da empresa, é necessária uma segurança centrada nos dados.

Pelo facto das aplicações de gestão documental e de processos geralmente trabalharem com todo o tipo de documentos de uma organização, a integração duma nova funcionalidade que permita proteger qualquer documento contido na aplicação, irá trazer mais garantias para todos os clientes que já utilizam, ou virão a utilizar, o iPortalDoc.

3.2 Resumo

Neste capítulo foram explicitados os problemas que motivaram a realização desta dissertação. Na primeira parte, o foco principal será na melhoria e correção de um problema do Contacts, que põe em causa a confidencialidade das entidades e contactos que se encontram na aplicação. Na segunda parte, o objetivo será aumentar a segurança da informação que está contida no iPortalDoc, criando uma solução de segurança centrada nos dados.

Capítulo 4

Arquitetura dos Sistemas e Solução

No capítulo 3 foram detalhados todos os problemas que motivaram a realização desta dissertação. No presente capítulo, começaremos por apresentar a estrutura do sistema e da aplicação para o Contacts e para o iPortalDoc, e, de seguida, irá ser detalhada a solução proposta para resolver os problemas apresentados no capítulo anterior.

4.1 Contacts

O Contacts é a aplicação *web* para gestão de entidades e contactos, desenvolvida pela IPBRICK SA.

Nesta secção, irá ser apresentada a arquitetura do sistema, o funcionamento da aplicação e a solução proposta para cada um dos problemas, acerca da confidencialidade, explicitados no capítulo anterior, referentes ao Contacts.

4.1.1 Arquitetura do Sistema

O Contacts é uma aplicação *web* e, como tal, é necessário um *browser* para aceder à mesma. São várias as ações desta aplicação que, quando executadas, acedem à base de dados através do servidor IPBRICK, onde está alojado o Contacts, e que retorna toda a informação que necessitam. A figura 4.1 representa a arquitetura do sistema do Contacts.

4.1.2 Arquitetura da Aplicação

O Contacts utiliza uma base de dados relacional PostgreSQL, onde armazena toda a informação associada. Essa mesma base de dados é utilizada não só pelo Contacts, como também pelo iPortalDoc.

A utilização desta aplicação permite gerir as entidades e contactos associados a uma organização. Desta forma, é possível armazenar para no futuro aceder a múltiplas informações de cada contacto, tais como a morada, o contacto telefónico e o e-mail. Na versão atual, o Contacts já

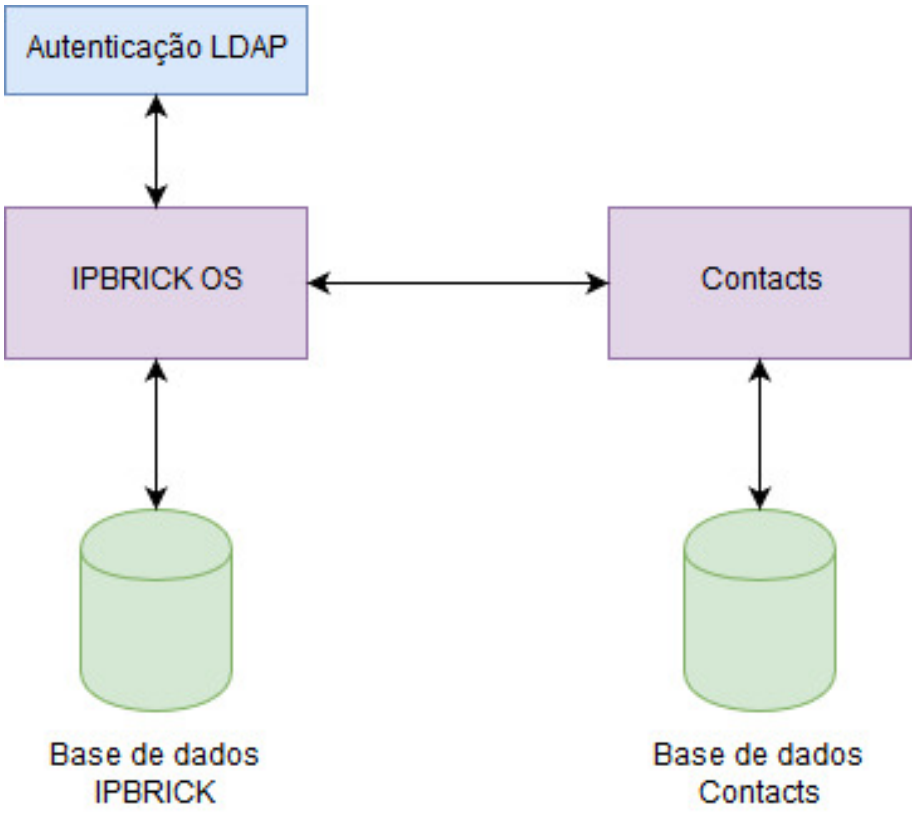

Figura 4.1: Arquitetura do sistema do Contacts

dispõe da possibilidade de atribuição de perfis de permissões a utilizadores desta aplicação, limitando o acesso à visualização ou edição dos dados das entidades e contactos. Esta funcionalidade é muito útil para organizações com vários projetos, pois permite atribuir a cada utilizador, ou a cada equipa, o nível de informação que deverá ter acesso.

Para começar a utilizar esta aplicação, em primeiro lugar é fundamental criar os utilizadores que irão utilizar a mesma. Mandatoriamente, a cada utilizador será atribuído um perfil de permissões, havendo por defeito três disponíveis: administrador, editor e leitor. Deverá existir sempre pelo menos um administrador, que tem permissões máximas de acesso, de edição e de criação. Após definir o perfil das permissões de acesso a cada utilizador, o passo seguinte é a criação de entidades e contactos. Ao criar uma entidade, é possível atribuir diversas características à mesma, sendo que o nome e o tipo de entidade a que pertence, são características de preenchimento obrigatório. Após a criação de uma entidade, passa a estar disponível a opção de criar contactos, que estarão associados à entidade. Após existirem utilizadores, entidades e contactos, o administrador da aplicação passa a poder atribuir permissões de acesso dos utilizadores a entidades e contactos, tendo sempre em conta o perfil de cada utilizador.

Tal como foi explicado no capítulo 3, existem três hipóteses de atribuir permissões aos utilizadores do Contacts. É possível associar utilizadores a tipos de entidade, associar entidades a utilizadores e associar utilizadores a entidades. Tendo sido associadas as permissões, cada utiliza-

dor só irá ter acesso às entidades que lhe forem permitidas pelo administrador e, dependendo do perfil, poderá visualizar, editar e/ou criar entidades e contactos na aplicação.

4.1.3 Solução

4.1.3.1 Permissões de Acesso a Entidades

O Contacts possibilita atribuir permissões de utilizadores a tipos de entidade e a entidades. Assim, para este problema (enunciado no capítulo 3) ser resolvido, há que ter em conta as duas tabelas da base de dados da aplicação, que nos informam os utilizadores que estão associados a tipos de entidade, e os utilizadores que estão associados a entidades individuais.

Na primeira fase do projeto, é necessário elaborar uma solução para que quando não existam utilizadores associados a um tipo de entidade e não existam utilizadores associados a nenhuma entidade desse mesmo tipo de entidade, todas as entidades e contactos pertencentes a este tipo passem a ser consideradas públicas. No momento em que todas as condições forem cumpridas, as entidades que serão consideradas como públicas, serão exibidas a todos os utilizadores, conforme o perfil atribuído a cada um. Isto é, um utilizador cujo perfil só permita visualizar entidades, só poderá fazer isso mesmo às entidades e contactos que agora tem acesso.

4.1.3.2 Confidencialidade das Entidades

Gerir permissões de um conjunto de utilizadores em relação a entidades no Contacts, pode ser uma tarefa complicada devido à falta de ferramentas para o fazer. Para resolver este problema, deverá ser implementada uma ferramenta de pesquisa que possibilite pesquisar todos os utilizadores que pertençam a uma entidade ou a um tipo de entidade.

O local apropriado para esta ferramenta será dentro do menu Permissões, no sub-menu Classificação (figura 4.2). Neste sub-menu, o administrador deve começar por escolher os utilizadores que deseja e de seguida poderá atribuir permissões de acesso a certos tipos de entidade, a entidades pertencentes a países ou a categorias selecionadas.

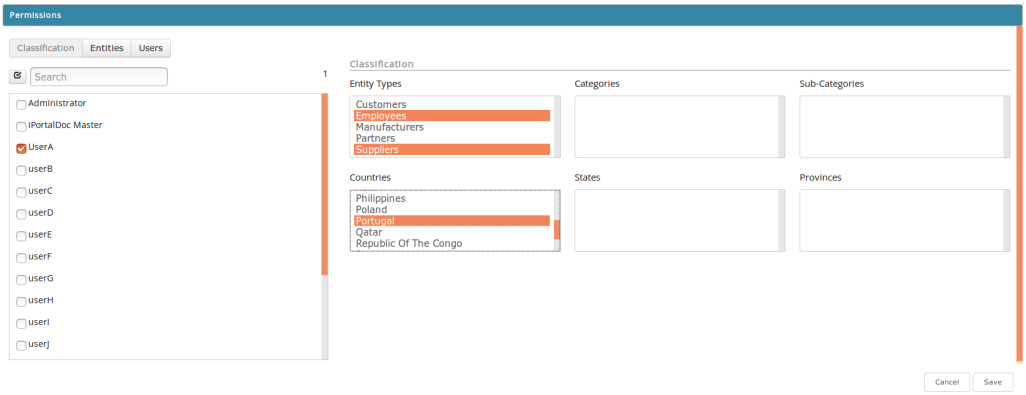

Figura 4.2: Atribuição de permissões por classificação

Para implementar esta ferramenta, irá ser necessária a utilização da tecnologia AJAX, pois desta forma irá ser feita a nova pesquisa dos utilizadores, sem atualizar a página. Deverão ser criados dois parâmetros de pesquisa: tipos de entidade e entidades. Ao utilizar este novo método de pesquisa de utilizadores da aplicação, haverão quatro possibilidades de ocorrência:

- Se selecionar um tipo de entidade e não selecionar uma entidade – apresentar todos os utilizadores com permissões de acesso ao tipo de entidade escolhido;
- Se selecionar uma entidade e não selecionar um tipo de entidade – apresentar todos os utilizadores com permissões de acesso à entidade escolhida;
- Se selecionar um tipo de entidade e selecionar uma entidade – apresentar a interseção dos utilizadores com permissões de acesso ao tipo de entidade e à entidade escolhida;
- Se não selecionar um tipo de entidade e não selecionar uma entidade – apresentar todos os utilizadores.

4.1.3.3 Conformidade com a Confidencialidade de Entidades e Contactos

Pelo facto do Contacts ter uma integração de informação com o LDAP do IPBRICK OS, é necessário que esta integração passe a ter em conta as permissões que estejam definidas na aplicação, para cada utilizador.

No estado atual, quando a sincronização dos dados para o LDAP está ativa, o Contacts já passa toda a informação. O problema é que é preciso filtrar essa informação que vem do Contacts, de forma a transmitir corretamente as permissões que cada utilizador tem na aplicação. Assim, quando um utilizador acede a uma aplicação como o IPBRICK.MAIL, esta vai fazer um pedido ao servidor LDAP, e o servidor vai enviar apenas as entidades e contactos que o utilizador tem permissões para visualizar, conforme está definido no Contacts.

Para solucionar este problema, iremos separá-lo em duas partes: a primeira parte irá dizer respeito ao trabalho a ser feito do lado do LDAP, e a segunda parte do lado do Contacts.

Em relação ao LDAP, deverão ser criadas entradas para cada utilizador do Contacts, com a indicação dos contactos e entidades que o utilizador pode aceder. Depois, cria-se uma lista de controlo de acessos (ACL) em que se o *id* da entidade ou contacto estiver na entrada do utilizador, este possa ver a informação dessa entidade/contacto.

No Contacts, é necessário verificar em que situações tem de ser modificada a informação nas novas entradas criadas no LDAP, tendo em conta as ações efetuadas no Contacts por parte dos utilizadores.

4.2 iPortalDoc

O iPortalDoc é uma aplicação *web* de gestão documental e de processos, desenvolvida pela IPBRICK SA.

Nesta secção, irá ser apresentada a arquitetura do sistema, o funcionamento da aplicação e a solução proposta para o problema apresentado, relativo à segurança da informação presente no iPortalDoc.

4.2.1 Arquitetura do Sistema

O iPortalDoc funciona em colaboração com o sistema operativo da IPBRICK, o IPBRICK OS. Além deste sistema operativo, o iPortalDoc precisa de uma base de dados onde possa armazenar os documentos. Também está integrado com um servidor web, um servidor de correio eletrónico, um servidor de ficheiros e, ainda, um servidor de informação e gestão de domínios [5]. Os utilizadores que podem ter acesso a esta aplicação são criados através da aplicação Contacts. A figura 4.3 mostra a arquitetura do iPortalDoc.

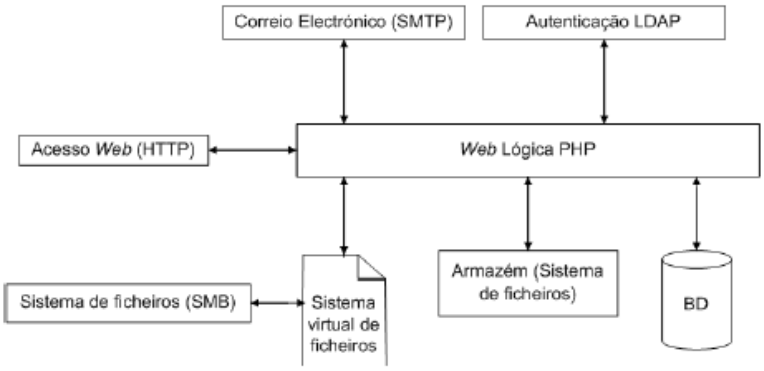

Figura 4.3: Arquitetura do iPortalDoc ([5])

4.2.2 Arquitetura da Aplicação

De forma semelhante ao Contacts, o iPortalDoc é também uma aplicação *web*, que utiliza uma base de dados relacional PostgreSQL, onde armazena toda a informação associada.

Esta aplicação criada pela IPBRICK SA, é um sistema de gestão documental e de processos, baseado em *workflows*, que funciona como um serviço de valor acrescentado para *Intranet*. Esta aplicação permite o registo, classificação e tratamento de todas as comunicações, bem como o tratamento processual de todos os processos de uma organização. A nível da gestão documental, o iPortalDoc disponibiliza diversas opções, tais como a introdução, o encaminhamento e a associação de documentos. A nível da gestão dos processos, a aplicação já disponibiliza alguns processos base, com *workflows* configurados e preparados para a utilização da maioria das empresas. Alguns dos processos mais utilizados são a gestão da correspondência, dos recursos humanos, financeira e comercial.

Na versão mais atual, esta aplicação disponibiliza funcionalidades relacionadas com permissões a utilizadores. Também, a qualquer momento de um determinado processo que decorra no iPortalDoc, haverá sempre acesso a todo o histórico de pessoas implicadas, intervenções realizadas, bem como documentos e emails associados.

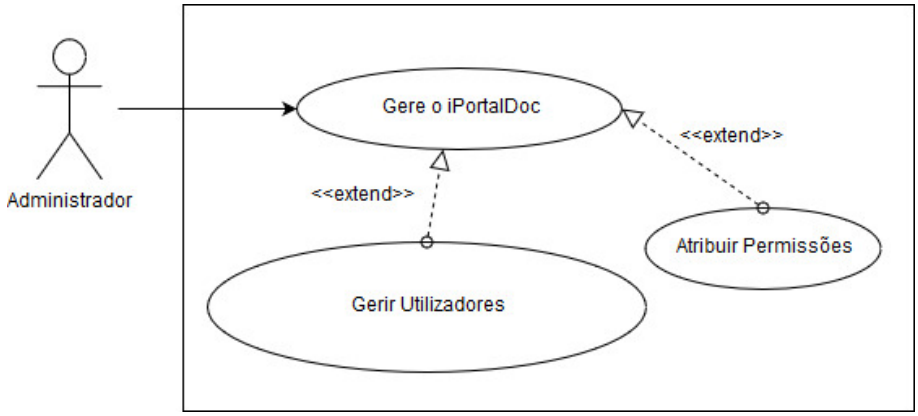

Figura 4.4: Diagrama de casos de uso de um administrador

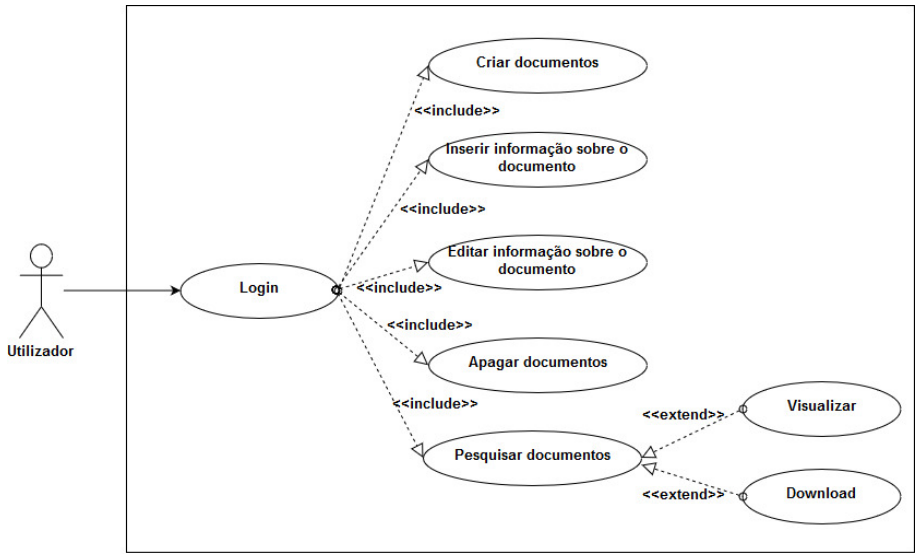

Figura 4.5: Diagrama de casos de uso de um utilizador comum

Na figura 4.4 e na figura 4.5 são apresentados os diagramas UML de casos de uso para um administrador da aplicação e para um utilizador comum, respetivamente.

4.2.3 Solução

Cada organização deve escolher o nível de segurança mais adequado para si, tendo em conta, por exemplo, a relação custo/beneficio. Empresas que utilizem o iPortalDoc para a gestão documental, necessitam que haja uma maior proteção de documentos, para evitar que sejam interceta-

dos por alguém que não deveria aceder aos mesmos. Então, para proteger ficheiros sensíveis que constem no iPortalDoc, irá ser aplicada uma segurança centrada nos dados.

Neste caso, o ideal seria a implementação de um método de prevenção de perda de dados (DLP) em conjunto com um método de proteção de dados. Com estes dois métodos implementados, o iPortalDoc tornar-se-ia muito mais seguro, pois estaria a controlar os dados que estão em uso, em movimento e em repouso.

Portanto, em primeiro lugar, deverá ser analisada uma solução DLP, mais especificamente, *Endpoint* DLP, pois assim será instalado um agente no final de cada sistema (nos computadores dos funcionários/utilizadores). Esta solução permite controlar a utilização de documentos de cada utilizador, impedindo que estes saiam da rede da empresa através de dispositivos USB, por e-mail ou através de um *upload* pela Internet.

Uma solução DLP é mais eficaz quando integrada com uma solução de encriptação. Como tal, o passo seguinte será selecionar um método de encriptação que seja capaz de proteger os documentos que estão no iPortalDoc. Existem duas situações em que esta ferramenta de encriptação deverá constar: aquando da criação de um documento e nas definições dos documentos que já estão presentes na aplicação. Como nem todos os documentos são considerados sensíveis, esta nova funcionalidade não irá obrigar a que todos os ficheiros sejam encriptados, fornecendo apenas a opção para o fazer. No caso de um utilizador criar um documento sem selecionar a opção encriptar, se no futuro esse documento passar a conter informações que só certos utilizadores devem ter acesso, haverá a opção de o encriptar e substituir a atual versão, na aplicação.

Capítulo 5

Implementação e Validação

Este capítulo corresponde à fase de implementação das soluções, para cada problema, apresentadas no capítulo 4. Em primeiro lugar, será feita uma introdução às ferramentas de apoio ao desenvolvimento desta dissertação. Seguidamente, irá ser detalhada toda a implementação desenvolvida para cada problema. Por fim, serão apresentados os resultados obtidos em cada fase, assim como a sua validação.

5.1 *Software* Utilizado no Desenvolvimento

Todas as implementações efetuadas ocorreram numa máquina virtual de testes da empresa. Foi utilizado o sistema operativo IPBRICK OS, na versão 6.3, baseado no Debian 7 (wheezy).

Relativamente às tecnologias de implementação, foi usado HTML, PHP, Javascript, Ajax e Bootstrap.

Durante toda a implementação, foram utilizadas ferramentas de *software*, das quais se destacam o Cervisia, o Sublime Text e o Terminal. O Cervisia é uma interface gráfica de um sistema que essencialmente, permite fazer o controlo de versões (torna possível trabalhar com várias versões de um ficheiro). O Sublime Text é um sofisticado editor de texto, com um bom desempenho, sendo muito utilizado por programadores, pois existem vários pacotes disponíveis, que fazem com que seja uma ferramenta muito útil para desenvolver código em qualquer linguagem de programação. Por fim, o Terminal é uma linha de comandos que, neste projeto, permitiu aceder à máquina virtual e à base de dados, permitiu fazer a verificação de *logs*, e fui utilizado para editar código referente ao LDAP (através do editor de texto Vi).

5.2 Contacts

Nesta secção, será apresentado todo o trabalho elaborado e implementado no Contacts, conforme os problemas que foram apresentados. A implementação das três fases seguintes, ocorreu na versão v5.0.10 do Contacts.

5.2.1 Permissões de Acesso a Entidades

Esta primeira fase de implementação tem como objetivo melhorar a forma como são atribuídas as permissões de acesso de utilizadores a entidades e contactos, presentes no Contacts.

Como já foi explicado anteriormente, cada utilizador pode estar associado a tipos de entidade e/ou apenas a entidades. Cada entidade tem, obrigatoriamente, pelo menos um tipo de entidade atribuído. Se um utilizador estiver associado a um tipo de entidade, significa que terá permissões para aceder ás entidades pertencentes a esse tipo, tendo em conta o perfil de utilizador que lhe foi atribuído.

O perfil de utilizador "Admin"nunca pode ser alterado ou eliminado, sendo que, os utilizadores com este perfil, visualizam toda a informação, independentemente das restantes configurações.

Se as permissões de um utilizador estiverem ativas, para serem apresentadas entidades ao mesmo, há que ter em conta os seguintes fatores:

- Perfil atribuído ao utilizador;
- Entrada desse utilizador na tabela *contacts_permissions_default* – tabela que indica o *id* dos utilizadores e os correspondentes tipos de entidade atribuídos;
- Entrada desse utilizador na tabela *ipcontactos_permissions* – tabela que indica o *id* dos utilizadores e as correspondentes entidades atribuídas;

Para além destes parâmetros, é necessário desenvolver a seguinte condição: se para um tipo de entidade não houver qualquer entrada na tabela *contacts_permissions_default*, e se para todas as entidades pertencentes a esse tipo não houver qualquer entrada na tabela *ipcontactos_permissions*, então isto significa que este tipo de entidade não estará sujeito a restrição de permissões, e como tal, todos os utilizadores do Contacts (tendo em conta o seu perfil), devem ter acesso ás entidades associadas a esse tipo de entidade.

A obtenção dos tipos de entidade a tornar público a todos os utilizadores, exige a interseção de dois conjuntos, que serão explicados seguidamente.

Uma vez que já existia, na base de dados do Contacts, uma tabela com os tipos de entidade que têm utilizadores associados, e outra com utilizadores associados a entidades individuais, em primeiro lugar, foram encontrados os tipos de entidade com entidades atribuídas. Em seguida, foram apurados os tipos de entidade com entidades, com utilizadores atribuídos. Aos tipos de entidade com entidades, foi feita a diferença dos tipos de entidade com entidades, com utilizadores, obtendo-se os tipos de entidade com entidades, sem utilizadores associados, sendo este o primeiro conjunto pretendido.

O segundo conjunto foi obtido através da diferença entre todos os tipos de entidade existentes e os tipos de entidade com utilizadores associados. Daí resulta o conjunto com os tipos de entidade sem utilizadores.

Por fim, foi feita a interseção entre os dois conjuntos, obtendo-se as entidades a tornar públicas a todos os utilizadores ativos do Contacts, tendo em conta o seu perfil. Toda esta implementação é verificada ao iniciar a aplicação.

5.2.1.1 Funções Implementadas

A implementação desta melhoria exigiu a criação e a utilização das seguintes funções, em *Javascript/Jquery*:

- *getIDTipoEntidades()* – Função que retorna o *id* de todos os tipos de entidade, exceto o "id = 1", que são os contactos privados. Esta função vai buscar os valores à tabela *tipoentidade* da base de dados. Dos dados que a tabela contém, foram necessários os seguintes: id (inteiro), nome (texto), ativo (boleano) e privado (boleano);

- *getIDTipoEntidadesComUsers()* – Função que retorna o *id* de todos os tipos de entidade, que têm utilizadores atribuídos. Esta função vai buscar os valores à tabela *contacts_permissions_default* da base de dados. Dos dados que a tabela contém, foram necessários os seguintes: idutilizador (inteiro) e *value* (texto). Destes, vale a pena explicar que o atributo *value* indica o *id* dos tipos de entidade de cada utilizador, sendo elas separadas por ":"no caso de um utilizador ter mais do que um tipo de entidade associado;

- *getIDUserTipoEntidadesComUsers()* – Função que retorna o *id* de todos os utilizadores com permissões para aceder a algum tipo de entidade. Esta função vai buscar os valores à tabela *contacts_permissions_default* da base de dados. Dos dados que a tabela contém, foram necessários os seguintes: idutilizador (inteiro) e *value* (texto);

- *getIDTipoEntidadesComEntidadesComUsers()* – Função que retorna o *id* de todos os tipos de entidade que contêm pelo menos uma entidade, no mínimo com um utilizador atribuído. Esta função vai buscar os valores à tabela *ipcontactos_permissions* e *entidade* da base de dados. Dos dados que a primeira tabela contém, foram necessários os seguintes: idutilizador (inteiro) e idatributo (inteiro); da segunda tabela, foram utilizados: identidade (inteiro) e idatributo (inteiro). Destes, vale a pena explicar que o atributo "idatributo"é o *id* do tipo de entidade;

- *getIDTipoEntidadesComEntidadesSemUsers()* – Função que retorna o *id* de todos os tipos de entidade, em que todas as entidades que fazem parte deste tipo, não contêm nenhum utilizador atribuído. Esta função vai buscar os valores à tabela *ipcontactos_permissions* e *entidade* da base de dados. Dos dados que a primeira tabela contém, foram necessários os seguintes: idutilizador (inteiro) e idatributo (inteiro); da segunda tabela, foram utilizados: identidade (inteiro) e idatributo (inteiro);

- *getIDTodosUsers()* – Função que retorna o *id* de todos os utilizadores ativos do Contacts. Esta função vai buscar os valores à tabela *utilizador_ipcontactos* da base de dados. Dos dados que a tabela contém, foram necessários os seguintes: idutilizador (inteiro) e nome (texto).

Tabela 5.1: Tabela com tipos de entidade e respetivas entidades e utilizadores associados

Tipos de entidade	Fabricante	Fornecedor	Parceiro	Cliente
Entidades	Entidade A	Entidade A Entidade B Entidade C	Entidade C	Entidade D Entidade E
Utilizadores associados ao tipo de entidade	Utilizador B	–	Utilizador D	–

5.2.1.2 Validação

De forma a validar a implementação, foram criados 4 tipos de entidade, 5 entidades com diferentes tipos de entidade atribuídos e 4 utilizadores com diferentes permissões de acesso. Na tabela 5.1 encontram-se os tipos de entidade criados, com as respetivas entidades pertencentes a cada um, e com a lista de utilizadores associados ao tipo de entidade. Na tabela 5.2 é possível observarmos os utilizadores que estão associados a cada entidade.

Das tabelas é possível observar que foram criadas 4 situações diferentes:

- Caso 1: Tipo de entidade (Fabricante) com utilizador e com entidade com utilizador;
- Caso 2: Tipo de entidade (Fornecedor) sem utilizador e com 2 entidades com utilizadores e outra entidade sem;
- Caso 3: Tipo de entidade (Parceiro) com utilizador e com entidade sem utilizador;
- Caso 4: Tipo de entidade (Cliente) sem utilizador e com entidade sem utilizador.

Com o trabalho que foi desenvolvido e implementado, seria de esperar que quando fosse atualizada a página principal do Contacts, que apenas as entidades D e E se tornassem públicas, ou seja, que passassem a ser visíveis a todos os utilizadores. Isto teria de acontecer pois é a única situação em que um tipo de entidade não contém utilizadores associados, e ao mesmo tempo, todas as entidades que fazem parte do mesmo, também não contêm nenhum utilizador associado.

Depois de atualizada a página da aplicação, aconteceu o que seria de esperar: todos os utilizadores (A, B, C e D) agora conseguem visualizar toda a informação das entidades D e E, que pertencem ao tipo de entidade Cliente.

5.2.2 Confidencialidade das Entidades

Após o desenvolvimento da melhoria no Contacts, referente ás permissões dos utilizadores em relação ás entidades que estes têm acesso, é agora necessário continuar a aumentar a confidencialidade das entidades na mesma aplicação.

Em primeiro lugar, foi analisado o menu "Permissões"do Contacts, acessível apenas a administradores da aplicação. Definiu-se que o local apropriado para implementar a ferramenta de

Tabela 5.2: Tabela com entidades e respetivos utilizadores associados

Entidades	Entidade A	Entidade B	Entidade C	Entidade D	Entidade E
Utilizadores associados à entidade	Utilizador A	Utilizador C	–	–	–

pesquisa seria no sub-menu "Classificação", que até ao momento apenas apresentava todos os utilizadores por ordem alfabética e só permitia selecionar (ou desselecionar) todos.

No sub-menu "Classificação", foi criado um botão de pesquisa (com o ícone de uma lupa) que, ao clicar, abre uma *popup* que permite selecionar um tipo de entidade e/ou uma entidade. Esta ferramenta desenvolvida pode ser visualizada na figura 5.1.

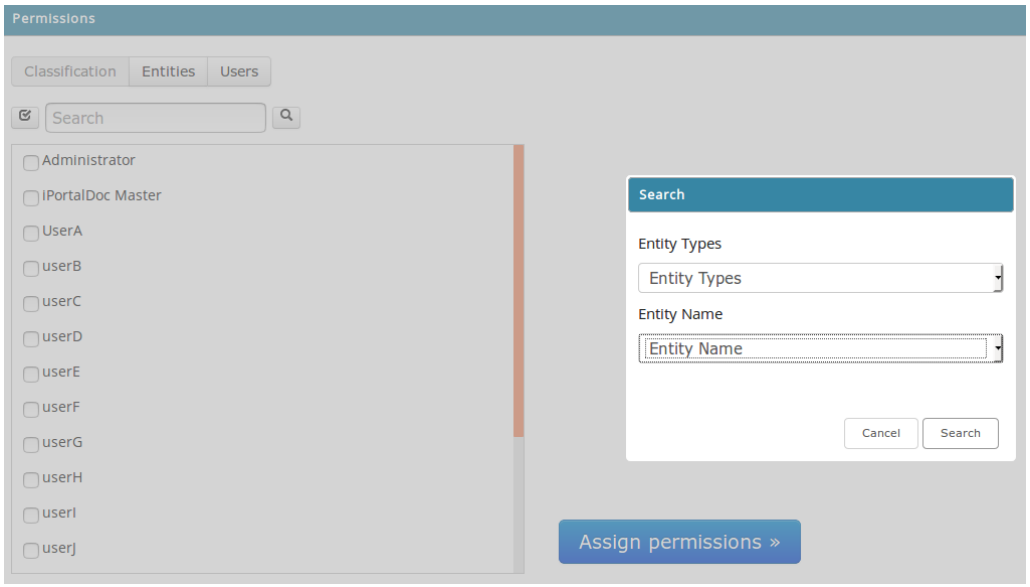

Figura 5.1: Interface da Ferramenta de Pesquisa

Toda a informação necessária da base de dados (os tipos de entidade, as entidades e os utilizadores) é carregada ao iniciar a página, pois esta nova funcionalidade foi implementada com recurso à tecnologia AJAX, que permite realizar pedidos ao servidor sem ter que atualizar a página. Ao selecionar um (ou mais) tipo(s) de entidade e clicar em pesquisar, a *popup* será automaticamente fechada e, do lado esquerdo, aparecerão apenas os utilizadores desse mesmo tipo, permitindo a sua fácil seleção para atribuição de permissões. Da mesma forma, é possível escolher entidades, apresentando apenas os utilizadores das mesmas, e também é possível escolher ambos (entidades e tipos de entidade), o que na pesquisa irá fazer a intersecção dos utilizadores pertencentes aos dois grupos.

5.2.2.1 Funções Implementadas

A implementação desta nova funcionalidade exigiu a criação e a utilização das seguintes funções, em *Javascript/Jquery*:

- *searchUsers()* – Esta função cria a *popup* quando o utilizador clica no botão de pesquisa implementado nesta fase. Executa um pedido de AJAX em que retorna e preenche, nos respetivos campos, todos os tipos de entidade e as entidades que o utilizador pode selecionar;

- *getPermissions()* – Esta função executa um pedido de AJAX com os dados selecionados por parte do utilizador, retornando a lista de utilizadores que fazem parte da pesquisa. No caso do utilizador clicar em cancelar, é fechada a *popup*, não executando nenhuma pesquisa. No caso do utilizador não escolher nem entidade nem tipo de entidade e clicar em pesquisar, esta função retorna todos os utilizadores do Contacts;

5.2.2.2 Validação

Em primeiro lugar, a validação desta fase passou por fazer todo o tipo de pesquisas possíveis. Começamos por selecionar apenas um tipo de entidade, e o resultado foram todos os utilizadores que pertencem a todas as entidades desse mesmo tipo e todos os utilizadores que estão diretamente associados ao tipo de entidade escolhido. De seguida, foi realizada uma pesquisa selecionando apenas uma entidade, do qual se obtiveram apenas os utilizadores com permissões para aceder a essa mesma entidade. Seguidamente, selecionou-se um tipo de entidade e uma entidade, da qual se obtiveram os utilizadores que estão associados ao tipo de entidade e, ao mesmo tempo, associados também à entidade selecionada. Por fim, não selecionando nem tipos de entidade nem entidade e clicando em pesquisar, obtiveram-se todos os utilizadores da aplicação, o que é o equivalente a fechar e voltar a abrir o menu "Permissões".

Após terem sido realizados estes primeiros testes, foram feitos novos testes em que se utilizaram entidades com mais do que um tipo de entidade atribuído, tipos de entidade com mais do que uma entidade, e utilizadores com permissões para aceder a apenas entidades, a apenas tipos de entidade ou a ambos. De todos estes, é de destacar o caso em que um utilizador tem mais do que um tipo de entidade atribuído: uma vez que a tabela *contacts_permissions_default*, da base de dados, tem o *id* dos tipos de entidade separados por ":", para cada utilizador, foi necessário separar estes *id's* para que a pesquisa funcionasse corretamente.

Os resultados de todos os testes efetuados foram verificados através de *query's* (equivalentes à pesquisa) na base de dados, antes de cada teste. A validação da implementação permitiu que esta nova funcionalidade do Contacts possa estar presente numa futura versão da aplicação.

5.2.3 Conformidade com a Confidencialidade de Entidades e Contactos

Tal como foi definido na solução apresentada no capítulo 4, este problema irá ser dividido em duas partes: a primeira relativa ao LDAP e a segunda ao Contacts.

No plano inicial, a ideia seria começar por criar uma entrada por cada utilizador do Contacts, em que se iriam colocar os contactos e entidades que o utilizador tem permissões para aceder. Isto faz sentido pois cada utilizador pode ter diferentes permissões para aceder a entidades/contactos. Após a análise desta solução, entendeu-se que existe um certo conjunto de utilizadores que têm as mesmas permissões e o mesmo perfil: os administradores. Ora, isto faz com que baste uma entrada no LDAP para todos os utilizadores com este perfil de administrador.

Assim, o primeiro passo foi criar uma entrada com o grupo de utilizadores com acesso total (os administradores), dentro da unidade organizacional (OU) Contacts. Esta entrada pode ser observada na figura 5.2, em que se denominou este grupo de utilizadores (nome comum) como "Todospodemver". É importante salientar que o "dc=ipdocdev91"e o "dc=net"referem-se à componente de domínio da máquina virtual onde esta implementação ocorreu. Para efeito de testes (que serão demonstrados adiante), definiu-se que os utilizadores que pertencem a este grupo são o "administrator"e o "userc".

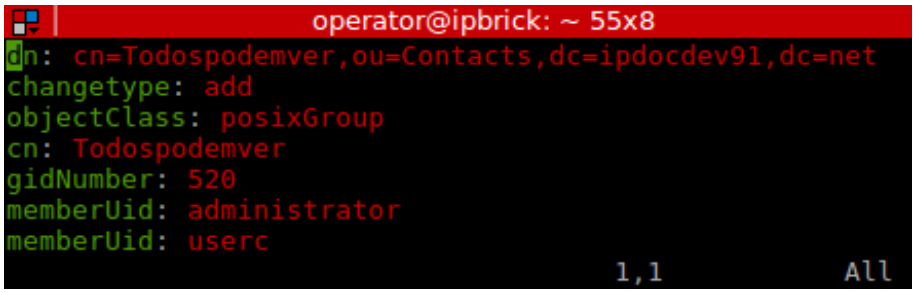

Figura 5.2: Entrada LDAP para administradores

O passo seguinte foi criar uma lista de controlo de acessos (ACL) para que estes utilizadores, com acesso total, possam ver todos os contactos e entidades.

Seguidamente, foi criada uma entrada por cada utilizador do Contacts, com a indicação dos contactos e entidades que o utilizador pode aceder. Esta entrada pode ser visualizada na figura 5.3, que exemplifica a entrada para o utilizador "userA". Neste exemplo, o utilizador apenas tem permissões para aceder à entidade com o *id*=2. Tal como no exemplo anterior, cada entrada tem um atributo "gidNumber", que identifica unicamente o grupo num domínio administrativo, tendo neste caso o valor 521.

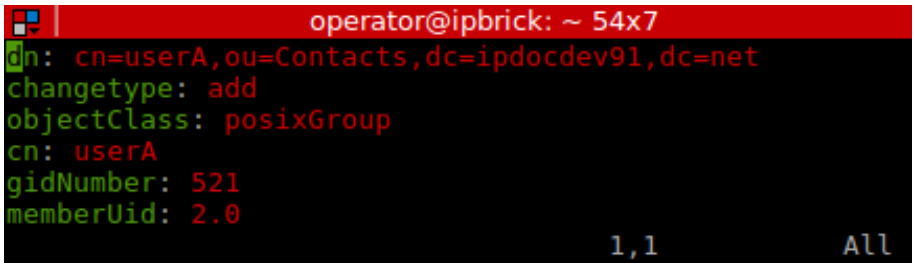

Figura 5.3: Entrada LDAP para um utilizador comum

Após ter sido criada uma entrada por cada utilizador, foi posteriormente criada uma ACL que faz com que se o *id* da entidade ou contacto estiver na entrada do utilizador, este possa visualizar a informação dessa entidade/contacto.

Tendo esta parte (do lado do LDAP) sido concluída, o passo seguinte seria avançar para a implementação do lado do Contacts. Visto que o tempo para a realização da dissertação é limitado, e que o objetivo principal deste problema era a elaboração de uma solução para o mesmo, esta fase do projeto não teve seguimento. Ainda assim, embora não tenha sido implementado o passo seguinte, foi elaborada uma solução para que alguém (no futuro) possa facilmente entender o que foi feito e o que será necessário fazer para completar e resolver o problema apresentado.

Então, do lado do Contacts, o próximo passo seria verificar em que situações tem de ser modificada a informação nas novas entradas criadas no LDAP, tendo em conta o comportamento ao:

- adicionar utilizadores;
- manipular permissões;
- criar novas entidades ou contactos;
- apagar entidades ou contactos;
- modificar o perfil do utilizador.

Ao adicionar utilizadores na aplicação, deverá ser criada uma entrada para cada um, do mesmo género da que foi apresentada na figura 5.3. Quando forem dadas permissões de entidades/contactos a um utilizador, terá que ser modificada a entrada inicial, passando o atributo *changetype* a ser *modify* em vez de *add*; também terá que ser adicionado o *id* da respetiva entidade/contacto que o utilizador teve acesso, bastando adicionar o atributo "*memberUid*: *id*". No caso de serem removidas permissões de um utilizador a uma entidade, basta alterar o atributo "*changetype*: *delete*".

5.2.3.1 Validação

De forma a validar o que foi implementado, foram utilizadas as entradas das figuras 5.2 (entrada para utilizadores com perfil administrador) e 5.3 (entrada para o utilizador "userA"). Foram feitas pesquisas no LDAP, utilizando o comando *ldapsearch*, através do Terminal. Desta forma, é possível simular o que aconteceria se um utilizador tentasse aceder ao *Groupware* da empresa, através de uma aplicação de e-mail.

Para efeito de testes, foram criados vários utilizadores com diferentes perfis, sendo que apenas o "administrator"e o "userc"fazem parte da entrada "Todospodemver"(figura 5.2), e apenas o "userA"tem uma entrada, que lhe dá acesso à entidade com o "id=2"(figura 5.3). Foram também criadas 4 entidades e 1 contacto (pertencente à entidade chamada "Entidade A").

Começou-se por verificar o caso em que um administrador, que pertence à entrada "Todospodemver", executa uma pesquisa das entidades e contactos existentes no Contacts. A pesquisa e o

resultado obtido, podem ser observados na figura A.1, em anexo. Tal como era de esperar, o utilizador "userc", que tem perfil de administrador (pertence à entrada "Todospodemver"), conseguiu ver os dados de todas as entidades e contactos do Contacts.

O próximo teste de validação foi fazer novamente uma pesquisa LDAP, mas desta vez pelo utilizador "userA", cuja entrada no LDAP lhe garante acesso apenas à entidade com "id=2". Na figura 5.4 encontra-se o resultado obtido da pesquisa pelo utilizador "userA". Observa-se que a única entidade que lhe foi possível visualizar foi a "IPBRICK. S.A.", que tem o "uid=2", o mesmo e único *id* que lhe foi concedido acesso através da entrada LDAP.

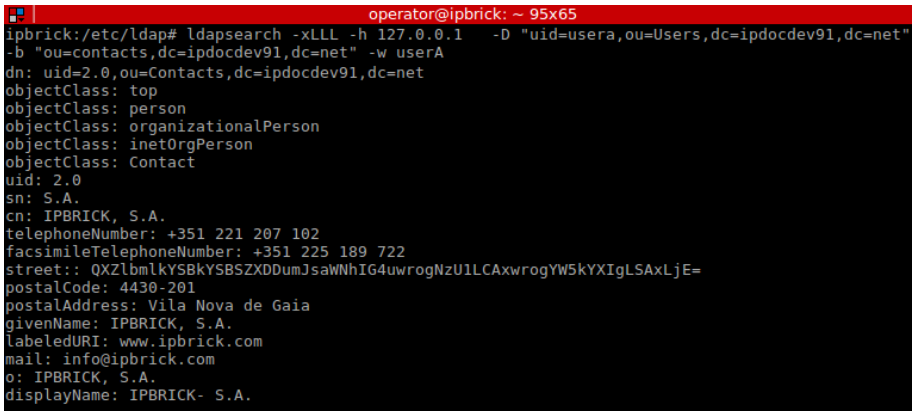

Figura 5.4: Pesquisa LDAP de utilizador com permissões restritas

Por fim, foi testado o caso de um utilizador do Contacts que não tem permissões de acesso a nenhuma entidade/contacto. A figura 5.5 comprova que um utilizador como o "userB", sem nenhuma entrada LDAP com as suas permissões de acesso, não tem acesso a nenhuma entidade nem contacto.

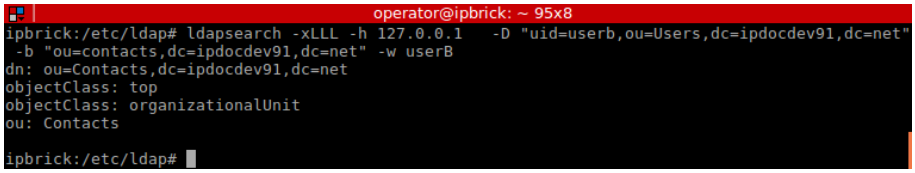

Figura 5.5: Pesquisa LDAP de utilizador sem permissões

Toda esta validação demonstra que as permissões do Contacts poderão agora ser transmitidas corretamente para o LDAP, e assim, quando outras aplicações, tais como o IPBRICK.MAIL, acederem ao LDAP, irão apresentar apenas as entidades/contactos conforme as permissões do utilizador no Contacts. Para isto acontecer, tal como já foi dito, é necessário tornar este processo automático, criando e modificando as entradas para cada utilizador, conforme as ações que este executa no Contacts.

5.3 iPortalDoc

Na presente secção, será apresentado o trabalho efetuado e implementado no iPortalDoc, conforme o problema apresentado. Foi utilizada a versão v6.2-beta2.15 do iPortalDoc.

Tal como foi indicado na solução proposta, para proteger dados sensíveis que constem em ficheiros armazenados no iPortalDoc, o ideal é implementar um método de prevenção de perda de dados (DLP), e, adicionalmente, implementar um método de proteção de dados. Por esta razão, a fase da implementação foi dividida em duas partes, que se encontram nas sub-secções seguintes.

5.3.1 Método de Prevenção de Perda de Dados

Tal como foi referido durante a caracterização do problema, para selecionar um método DLP, é necessário cumprir um requisito imposto pela empresa: o *software* deverá ser gratuito e *open-source*. Após a investigação das aplicações existentes no mercado que cumprissem este requisito, entendeu-se que a mais apropriada para o iPortalDoc seria o MyDLP.

MyDLP é uma solução DLP que originalmente era inteiramente gratuita e *open-source*. Mas, desde 2014, quando a empresa foi comprada pelo Comodo Group, esta deixou de dar suporte à aplicação gratuita, existindo agora apenas a última versão *open-source* de 2014. Para além desta, a empresa continua a disponibilizar uma versão paga da aplicação.

Ainda assim, tendo em conta o facto da escolha de soluções deste género (com as características desejadas) ser muito limitada, o MyDLP é uma aplicação com diversas vantagens. Esta aplicação é composta por 3 componentes: MyDLP Network, MyDLP Endpoint e MyDLP Web UI. O MyDLP Network é o servidor de rede que é responsável por controlar as operações da rede de uma empresa, tais como intercetar conexões TCP/IP e controlar o tráfego em aplicações *web*. O MyDLP Endpoint é um agente que é instalado em cada fim de sistema, possibilitando inspecionar as operações dos utilizadores, tais como imprimir documentos ou copiar ficheiros para dispositivos externos. Por fim, o MyDLP Web UI é uma interface para os administradores do sistema, que permite fazer toda a configuração do MyDLP Network e Endpoint.

A utilização deste *software* irá permitir controlar dados em movimento e em repouso. Através da interface de configuração, será possível definir novas regras ou utilizar regras que já existem por defeito, podendo estas proibir a saída de ficheiros que contenham dados que sejam considerados sensíveis. Por exemplo, no caso específico do iPortalDoc, se um administrador quiser impedir que documentos armazenados no iPortalDoc, que contenham contas bancárias ou números de cartões de crédito, saiam da rede da empresa, basta criar uma regra para tal. Para isso, defini-se na interface a fonte (o iPortalDoc), o destinatário (o domínio), o tipo de informação (contas bancárias e números de cartões de crédito) e o tipo de ação a executar (permitir, bloquear, registar ou colocar em quarentena). Ao definir a regra, também é possível que o administrador seja notificado por e-mail no caso desta regra ser executada. Neste exemplo, se um utilizador tentar enviar por e-mail um ficheiro que não era suposto, irá receber uma notificação por e-mail, de que tentou enviar informação considerada sensível, sendo que irá ser negado o envio do ficheiro.

Este método permite barrar a saída de informações sensíveis, controlando:

- a cópia dos dados para dispositivos externos: dados transferidos de *endpoints* são interceta-dos e inspecionados;
- a tentativa de imprimir o documento que contém os dados sensíveis: o processo de impres-são será intercetado e o documento será inspecionado;
- *print-screens* aos dados: serão controlados os *print-screens* feitos nos *endpoints*.

Na figura 5.6 encontra-se o esquema da localização do servidor MyDLP e de exemplos onde o MyDLP Endpoint deverá ser instalado. O MyDLP deverá ser integrado com [6]:

- Servidor de e-mail da IPBRICK, para proteger o tráfego SMTP;
- Servidor de diretórios, para utilizar o diretório de utilizadores e de grupos, durante as regras;
- Servidor proxy, para intercetar o tráfego *web* e protegê-lo.

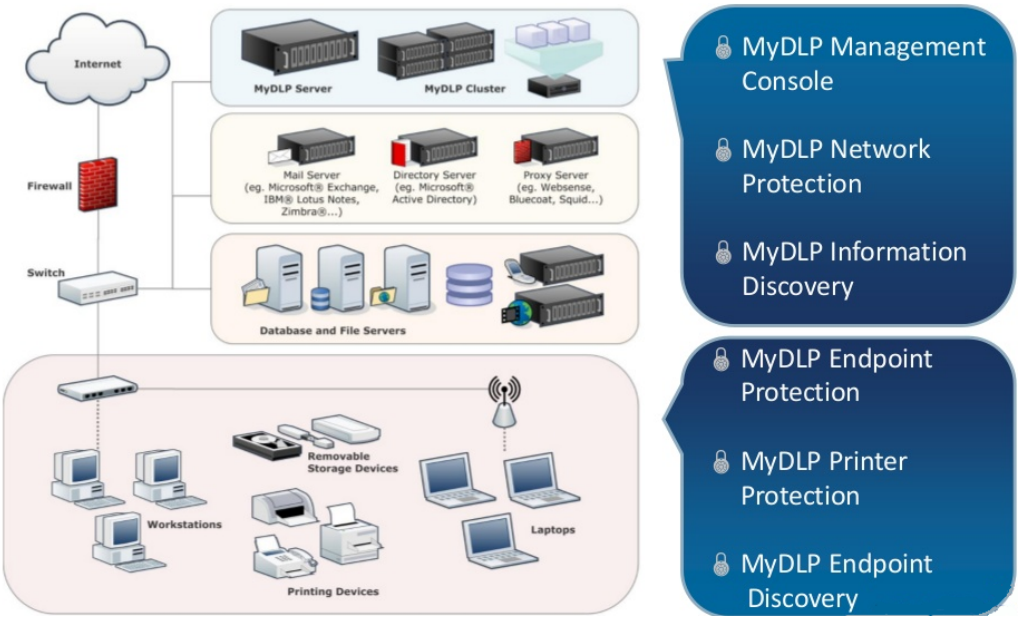

Figura 5.6: Esquema MyDLP de integração numa organização [6]

Após ter sido feita a análise do MyDLP, o passo seguinte seria a implementação desta solução na IPBRICK SA, de forma a poder gerir a segurança de dados sensíveis que se encontram no iPortalDoc. Para isso, em primeiro lugar seria necessário instalar o servidor MyDLP. Porém, devido ao facto da infraestrutura de rede da IPBRICK ser bastante distribuída, a configuração correta deste serviço seria uma tarefa que se revelou muito difícil. Por essa razão, não foi possível instalar o servidor MyDLP e, por conseguinte, não foi possível avançar com a implementação.

Embora não tenha sido exequível a implementação do MyDLP na empresa, a solução proposta seria o ideal para o problema em causa, pois tem todas as funcionalidades que eram pretendidas.

Comparando esta solução (gratuita) com a solução (paga) da Symantec (introduzida no capítulo 2), as desvantagens não são muito relevantes para o que é necessário para este projeto. Essencialmente, a Symantec [23] tem as seguintes vantagens, comparativamente com o MyDLP:

- Deteção de dados sensíveis, através de um sistema de reconhecimento de imagem;
- Sistema de *machine learning*: aprende e identifica dados sensíveis não estruturados, que são complicados de ser descritos, tais como documentos financeiros;
- *Symantec Information Centric Encryption*: protege dados sensíveis em aplicações *cloud*. Ficheiros sensíveis podem ser automaticamente protegidos através de encriptação. O acesso aos dados é controlado através da identificação do utilizador, e o perfil das permissões também poderá ser limitado com esta solução;
- Atualizações (melhorias) de *software* ao longo do tempo.

5.3.2 Método de Proteção de Dados

Nesta fase do projeto, foram analisadas soluções de proteção de dados, gratuitas e *open-source*, que pudessem ser implementadas no iPortalDoc. Concluiu-se a a solução mais apropriada é o GnuPG.

GnuPG, também conhecido como GPG, é uma implementação *open-source* da solução comercial PGP. Este *software* de encriptação é baseado no *standard* OpenPGP. Através do GnuPG é possível encriptar e desencriptar dados ou ficheiros.

No capítulo 2 encontra-se uma breve explicação deste método. Agora, tendo em atenção o problema específico desta dissertação, o primeiro passo para a implementação deste *software* é a instalação do mesmo; mas, para quem utiliza o Debian (como é o caso da IPBRICK), este *software* já vem instalado por defeito. De seguida, é preciso criar a chave pública e privada, para cada utilizador do iPortalDoc. Para isso, após se ter digitado o comando de geração de chaves no Terminal, será necessário escolher o tipo de algoritmo a ser utilizado para a geração das chaves. É possível escolher um dos seguintes algoritmos:

- RSA e RSA (*default*): é um dos algoritmos de encriptação assimétrica, mais utilizados. O GnuPG suporta tamanhos de chave entre os 1024 e os 4096 bits, para este algoritmo;
- DSA e Elgamal: este algoritmo requer um alto nível de entropia para a criação da chave. De forma semelhante ao algoritmo RSA, o GnuPG também suporta tamanhos de chave compreendidos entre os 1024 e os 4096 bits, para este algoritmo;
- DSA (assinatura apenas): tal como no algoritmo anterior, este também requer um elevado nível de entropia, mas desta vez apenas para a criação da assinatura. Esta opção permite assinar um ficheiro, sem o encriptar;
- RSA (assinatura apenas): este algoritmo funciona de forma semelhante ao DSA (assinatura apenas), sendo que as assinaturas são um pouco maiores em relação ao DSA.

Durante a primeira geração de chaves, foi escolhido algoritmo RSA e RSA, com um comprimento de 2048 bits. Seguidamente, é pedido para selecionar a validade da chave, ou seja, escolher o número de dias para a chave ser válida. Para efeitos de teste, foi escolhido que a chave não expira.

Ainda durante a mesma fase, foi então pedido para criar um utilizador. Foi digitado um nome, um email, um comentário (opcional) e uma *password*. Depois destes passos, é então que o GnuPG gera as chaves para o utilizador criado. O programa necessita de gerar muitos *bytes* aleatórios, e para o fazer, é uma boa ideia executar outras ações no computador (por exemplo, mexer o rato ou digitar no teclado), para que o gerador de números aleatórios tenha uma maior hipótese de ganhar entropia suficiente. Depois de estarem disponíveis *bytes* aleatórios suficientes, as chaves serão geradas. Para testar este método, foram criados os seguintes dados:

- Utilizador: Bruno;
- E-mail: bruno@email.pt.

Com estes dados, obtiveram-se os seguintes resultados, com o GnuPG:

- Chave pública (pub): 2048R/27CBFEE5E2110E4E;
- *Key ID*: 27CBFEE5E2110E4E.

Agora que já foi executado todo o processo da criação de chaves para o utilizador "Bruno", é necessário exportar a chave pública. Ou seja, temos de publicar a chave pública num servidor de chaves, para que outros utilizadores possam utilizar essa chave pública para enviar ficheiros ao utilizador "Bruno". Para isso, utilizamos um comando próprio do GnuPG, onde é exportada a chave pública do utilizador, para um novo ficheiro. Assim, agora já podemos partilhar a chave pública, através desse ficheiro, enviando-o para outros utilizadores que queiramos.

Após ter sido criada esta chave para o utilizador, o mesmo processo foi repetido, mas para um novo utilizador, chamado "Tiago", com o email "tiago@email.pt". Existindo agora dois utilizadores, será feito o primeiro teste de encriptação, que consistirá no envio de um ficheiro encriptado pelo "Bruno", para apenas o "Tiago". Em primeiro lugar, é preciso que a chave pública do "Tiago"pertença ao *key ring* do "Bruno". Após executar esse comando, está na altura de encriptar um ficheiro. Na figura 5.7 encontra-se o conteúdo de um ficheiro chamado "testeA.txt". A ideia é utilizar o GnuPG para encriptar este ficheiro, e depois desencriptar esse novo ficheiro obtido, de forma a verificar se o ficheiro final é igual ao ficheiro original.

Assim, foi utilizado, no Terminal, o comando "gpg -e -r "tiago@email.pt"testeA.txt". Este comando significa encriptar (-e) com GPG, sendo o utilizador com o email "tiago@email.pt"o recipiente (-r), para o ficheiro testeA.txt. Neste comando também se poderia ter identificado quem é o utilizador que encriptou o ficheiro, para que o utilizador que recebe o ficheiro encriptado, possa saber qual a origem do documento. O resultado será a criação dum ficheiro encriptado, com o nome "testeA.txt.gpg". Abrindo o ficheiro encriptado com um editor de texto, a mensagem que irá ser visualizada não será nada parecida com a original, tal como podemos observar na figura 5.8.

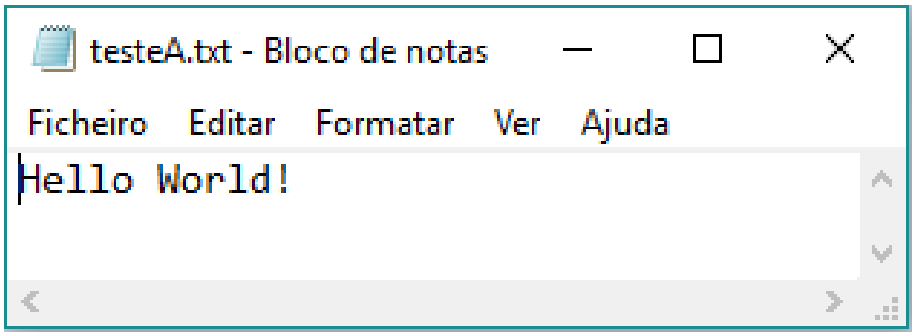

Figura 5.7: Ficheiro original

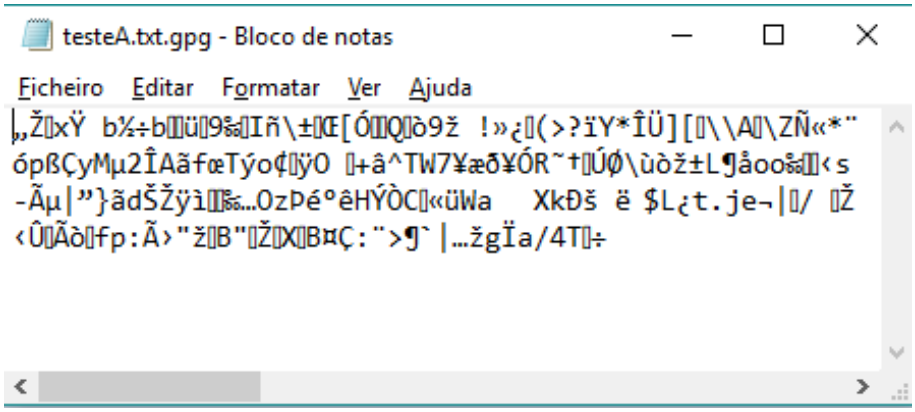

Figura 5.8: Ficheiro encriptado

Para verificarmos se este método realmente funciona, desencriptámos-lo através do GnuPG. Para isso, introduzimos a *password* do recipiente do ficheiro (utilizador "Tiago"). O resultado obtido é um novo ficheiro, com o conteúdo e extensão exatamente iguais ao original, da figura 5.7.

Tendo sido feita a análise e verificação do GnuPG, o passo seguinte é implementar este método no iPortalDoc. Visto que a tecnologia apropriada para a implementação na aplicação *web* é o PHP, o plano ideal seria a utilização funções pré-definidas de uma biblioteca PHP. No *website* oficial do PHP encontra-se uma expansão chamada "GnuPG", que consiste num vasto conjunto de funções que permitem encriptar, desencriptar, adicionar e remover chaves, verificar assinaturas, etc.

Sendo que esta expansão tem as funções pretendidas, tentamos instalar na máquina virtual. O problema é que não foi possível instalar a expansão, devido a problemas relativos a dependências. Mais especificamente, o erro obtido dizia *"fop-0.95 is not installed... pre-dependency problem - not installing system-doc"*. Após várias tentativas para resolver o erro, entendemos que este estava diretamente relacionado com o facto da versão 7 do Debian (instalada na máquina virtual) ser demasiado antiga para instalar e utilizar esta expansão (a versão atual disponível é a Debian 9). Visto que o sistema operativo IPBRICK, instalado na máquina virtual, é baseado em Debian 7, não foi possível avançar com a solução, por este caminho.

Para resolver este problema, sem ter que utilizar a expansão GnuPG para PHP, foi analisada

outra alternativa. A nova solução seria utilizar a função PHP `shell_exec()` para executar as funções GnuPG. A função `shell_exec()` permite colocar dentro dela, o comando que se colocaria no Terminal, executando-o da mesma forma no decorrer da *script* implementada. Desta forma, iriam ser utilizadas as mesmas funções das quais fizemos as verificações iniciais do GnuPG.

De forma a testar esta solução, começamos por criar uma simples *script* PHP, que consistia na apresentação duma página *web* simples, com um botão, que quando clicado iria encriptar um ficheiro que estava na própria máquina virtual. Ao testar esta solução, descobrimos que esta não funcionou, não tendo encriptado o ficheiro. Para tentar descobrir qual seria o problema, começamos por testar a função `shell_exec()`, executando outro comando no terminal: criando um ficheiro. Ao fazer este teste, todo o código foi executado corretamente, e o ficheiro foi criado, comprovando que o problema não seria da função `shell_exec()`. Após várias pesquisas, foi possível entender que o problema surge da tentativa de executar uma função GnuPG através da função `shell_exec()` no *browser*, sendo que podemos concluir que não o é possível fazer, não havendo solução para este caminho. De forma a ter a certeza que é o *browser* que impede que qualquer comando *gpg* seja executado, foi criada uma *script* PHP que apenas executava um `shell_exec()` para encriptar um ficheiro existente na máquina virtual. Ao executar diretamente através do Terminal (sem utilizar o *browser*), comprovamos que assim funciona corretamente, tendo encriptado o ficheiro. Logo, esta segunda solução não é válida e não poderá ser utilizada.

Sendo o tempo da realização da dissertação limitado, não foi possível avançar com a implementação de um método de proteção de dados, restando agora tirar conclusões do que foi desenvolvido. Após estas duas diferentes tentativas de implementação do GnuPG, é possível concluir que a primeira será o ideal a utilizar num trabalho futuro, quando for lançado o novo IPBRICK OS, baseado em Debian 9, que de momento já se encontra em fase de testes pela empresa. Assim, já será possível instalar a expansão GnuPG para PHP, e tirar partido de todas as funções que poderão ser úteis para a implementação deste método.

Ainda assim, de forma a pensar no trabalho futuro para este projeto, foram desenvolvidas as interfaces no iPortalDoc, para implementar este método de encriptação. Na figura 5.9 encontra-se o detalhe das informações associadas ao documento chamado "documento A", armazenado no iPortalDoc. Do lado direito da figura 5.9, estão situadas as opções disponíveis para executar neste documento. Destas, foram implementadas as opções *Encrypt* e *Decrypt*.

Na figura A.2 encontra-se a interface para a encriptação de um ficheiro que esteja no armazém do iPortalDoc. A ideia será que o utilizador, que pretende que o ficheiro seja apenas visível para certos utilizadores, digite o email do(s) recipiente(s), que terão obrigatoriamente de fazer parte do seu *key ring*. Depois basta clicar em *Encrypt*, e o ficheiro será encriptado e atualizado no iPortalDoc: será eliminado o ficheiro atual e será feito o *upload* deste.

Na figura A.3 encontra-se a interface para a desencriptação de um ficheiro que esteja no armazém do iPortalDoc. Basta o utilizador clicar na opção de desencriptar, em que depois lhe irá ser pedido para inserir a sua *password*, que caso seja corretamente digitada, o ficheiro será desencriptado e iniciará o seu *download*.

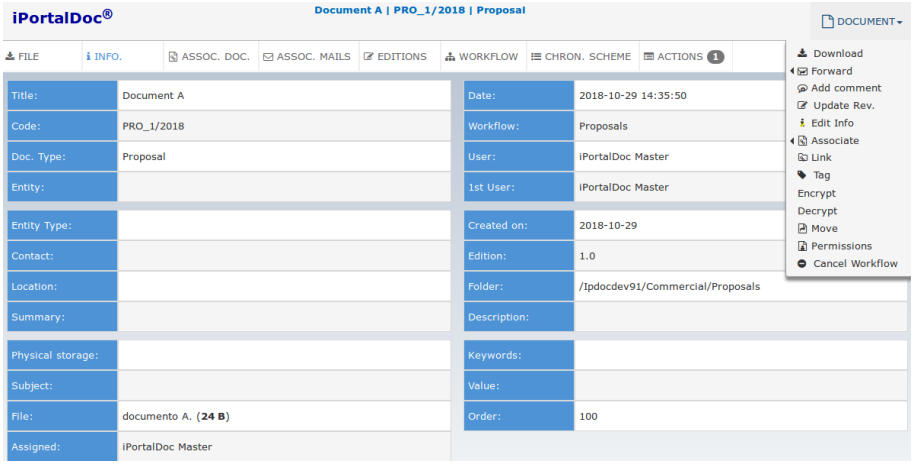

Figura 5.9: Interface para implementação da encriptação e desencriptação

5.4 Resumo

Neste capítulo, começamos por fazer uma introdução acerca do *software* que foi utilizado durante o desenvolvimento da dissertação. A fase da implementação foi dividida em duas partes, sendo primeiro relativo ao Contacts e depois ao iPortalDoc.

Em relação ao Contacts, foram implementadas duas melhorias na aplicação e foi elaborada uma solução para um problema de confidencialidade das entidades e contactos. Essencialmente, o primeiro problema consistiu na implementação de uma melhoria que fizesse com que tipos de entidade, sem nenhum utilizador associado, se tornassem visíveis a todos os utilizadores da aplicação, fazendo com que as entidades pertencentes a esse tipo sejam automaticamente associadas a todos os utilizadores, conforme o perfil de cada um. A implementação da solução, relativa ao segundo problema, foi a criação de uma nova ferramenta de pesquisa de utilizadores, que permite filtrar a pesquisa através da escolha de entidades ou de tipos de entidade. Esta funcionalidade passa agora a ser uma ferramenta muito útil para o Contacts, pois permite reduzir largamente o tempo de atribuição de permissões a utilizadores, e ainda reduz a probabilidade de errar na fase de atribuição de permissões. Relativamente ao último problema do Contacts, foram implementadas listas de controlo de acessos que, quando combinadas com entradas de cada utilizador, permitem corrigir o problema e enviar corretamente para o servidor LDAP, as permissões de acesso para cada utilizador, conforme o que é definido no Contacts. Nesta última fase, fica a faltar a implementação do lado do Contacts, que deverá tornar todo o processo automático (a criação e a modificação das entradas no LDAP).

A elaboração e implementação da solução relativa ao iPortalDoc, foi dividida em duas partes: método de prevenção de perda de dados e método de proteção de dados. O método de prevenção de perda de dados escolhido foi o MyDLP, que é um *software* gratuito e *open-source*. Foi realizada uma análise aos requisitos e foram explicadas as principais funcionalidades. No final, entendeu-se que esta solução seria complicada para implementar numa empresa como a IPBRICK, que tem

uma infraestrutura de rede muito distribuída. Avançamos para o método de proteção de dados, em que foi escolhido o GnuPG para ser implementado no iPortalDoc. O GnuPG permite encriptar documentos através de alguns dos algoritmos mais utilizados no mundo. Foi feita a verificação deste método e depois tentou-se implementá-lo, sendo que não foi possível, devido à versão do Debian ser demasiado antiga para utilizar uma expansão PHP adequada para o uso de GnuPG. Por fim, foram ainda desenvolvidas interfaces que poderão ser utilizadas para a implementação do método de proteção de dados, quando for lançada uma versão mais atual do IPBRICK OS, que seja baseada em Debian 9.

Capítulo 6

Conclusões e Trabalho Futuro

Após ter sido apresentado todo o trabalho desenvolvido nesta dissertação, resta agora apresentar as conclusões e sugerir trabalho a ser realizado no futuro.

6.1 Conclusão

A dissertação foi dividida em duas partes. O objetivo da primeira parte foi aumentar a confidencialidade das entidades e contactos, implementando melhorias no Contacts e corrigindo um problema referente à transmissão de permissões dos utilizadores para o LDAP. O objetivo da segunda parte da dissertação foi aumentar a segurança da informação contida no iPortalDoc, elaborando soluções de métodos de proteção e de prevenção de perda de dados.

Relativamente ao Contacts, as duas melhorias que foram implementadas garantem um redução drástica do tempo que até ao momento era necessário despender para fazer as mesmas tarefas. Para além disso, pelo facto destes dois processos se tornarem muito mais simples para organizações com muitos utilizadores, haverá naturalmente uma diminuição dos erros de atribuição de permissões, por parte dos administradores.

Em relação ao problema de transferência de permissões dos utilizadores do Contacts para o servidor LDAP, foi realizada apenas a implementação da solução do lado do LDAP. Embora esta fase da dissertação não tenha sido totalmente desenvolvida, foram implementadas as listas de controlo de acesso e algumas entradas LDAP dos utilizadores, que provam que basta tornar todo o processo automático (implementar do lado do Contacts) para que o problema seja plenamente resolvido. Com a solução desenvolvida, a confidencialidade das entidades e contactos deixará de estar em risco de não cumprir com o estipulado no RGPD (regulamento que entrou em vigor em 2018).

Na segunda parte da dissertação, foram elaboradas as soluções de dois métodos que, idealmente, deveriam ser implementados no iPortalDoc para aumentar a segurança dos dados. O método de prevenção de perda de dados analisado foi o MyDLP, e concluiu-se que, embora tivesse todas as funcionalidades que seriam ideais para o problema em questão, não foi possível integrar na IPBRICK devido à complexidade da sua infraestrutura de rede. Logo, métodos como o MyDLP,

são mais adequados a organizações com infraestruturas de rede mais simples, ou então, devem ser implementados no início da criação das infraestruturas. O método de proteção de dados analisado foi o GnuPG, que disponibiliza alguns dos algoritmos mais utilizados e mais seguros, para a encriptação de ficheiros. Foi verificado o GnuPG, e concluiu-se que não foi possível utilizá-lo devido à incompatibilidade com a versão do sistema operativo disponibilizado, na máquina virtual. Este problema será facilmente ultrapassado quando for lançada a nova versão do IPBRICK OS, que se encontra numa fase de desenvolvimento avançada. A implementação deste método no iPortalDoc iria criar uma nova funcionalidade no iPortalDoc, que faria com que os utilizadores tivessem a escolha de proteger ficheiros que pudessem conter informações sensíveis.

6.2 Trabalho Futuro

Tal como já foi explicado, não foi possível implementar tudo o foi proposto para esta dissertação, devido a restrições temporais e a contratempos que não foram detetados atempadamente. Contudo, tendo sido desenvolvidas as principais fases de cada problema, resta agora deixar sugestões a seguir para trabalho futuro.

Em relação ao Contacts, será necessário automatizar o processo de criação de entradas LDAP, para que as aplicações como o IPBRICK.MAIL apresentem as informações das entidades e contactos aos utilizadores, coerentemente com as permissões indicadas no Contacts.

Sendo que a infraestrutura da IPBRICK SA é bastante complexa, caso a empresa pretenda realmente que seja implementado um método de prevenção de perda de dados, deverão ser analisadas outras aplicações no mercado, sendo que deverão estar abertos à possibilidade de que seja necessário uma aplicação paga.

Por último, visto que o GnuPG é uma das melhores soluções no mercado com as características pretendidas para o iPortalDoc e pela empresa (é gratuito e *open-source*), quando for lançada a nova versão do IPBRICK OS, baseada em Debian 9, já poderá será implementado este mecanismo com facilidade, segundo os resultados obtidos nesta dissertação.

Anexo A

Anexos

A.1 Pesquisa LDAP

A.2 Interface Encriptação

A.3 Interface Desencriptação

```
operator@ipbrick: ~ 96x65
ipbrick:/etc/ldap# ldapsearch -xLLL -h 127.0.0.1   -D "uid=userc,ou=Users,dc=ipdocdev91,dc=net"
-b "ou=contacts,dc=ipdocdev91,dc=net" -w userC
dn: ou=Contacts,dc=ipdocdev91,dc=net
objectClass: top
objectClass: organizationalUnit
ou: Contacts

dn: uid=4.0,ou=Contacts,dc=ipdocdev91,dc=net
objectClass: top
objectClass: person
objectClass: organizationalPerson
objectClass: inetOrgPerson
objectClass: Contact
uid: 4.0
sn: A
cn: Entidade A
telephoneNumber: 111
facsimileTelephoneNumber: 333
street: Address
postalCode: postal code
postalAddress: city
givenName: Entidade A
mail: email@ipbrick.com
mobile: 222
o: Entidade A
displayName: Entidade A

dn: uid=5.4,ou=Contacts,dc=ipdocdev91,dc=net
objectClass: top
objectClass: person
objectClass: organizationalPerson
objectClass: inetOrgPerson
objectClass: Contact
uid: 5.4
sn: A
cn: Contact A
givenName: Contact A
o: Entidade A
displayName: Contact A

dn: uid=10.0,ou=Contacts,dc=ipdocdev91,dc=net
objectClass: top
objectClass: person
objectClass: organizationalPerson
objectClass: inetOrgPerson
objectClass: Contact
uid: 10.0
sn: B
cn: Entidade B
telephoneNumber: 9192939495
street: Address B
givenName: Entidade B
o: Entidade B
displayName: Entidade B

dn: uid=2.0,ou=Contacts,dc=ipdocdev91,dc=net
objectClass: top
objectClass: person
objectClass: organizationalPerson
objectClass: inetOrgPerson
objectClass: Contact
uid: 2.0
sn: S.A.
cn: IPBRICK, S.A.
telephoneNumber: +351 221 207 102
facsimileTelephoneNumber: +351 225 189 722
street:: QXZlbmlkYSBkYSBSZXDDumJsaWNhIG4uwrogNzU1LCAxwrogYW5kYXIgLSAxLjE=
postalCode: 4430-201
postalAddress: Vila Nova de Gaia
givenName: IPBRICK, S.A.
labeledURI: www.ipbrick.com
mail: info@ipbrick.com
o: IPBRICK, S.A.
displayName: IPBRICK- S.A.

dn: uid=12.0,ou=Contacts,dc=ipdocdev91,dc=net
objectClass: top
objectClass: person
objectClass: organizationalPerson
objectClass: inetOrgPerson
objectClass: Contact
uid: 12.0
sn: C
cn: Entidade C
telephoneNumber: 123123123
street: Vila do Conde
givenName: Entidade C
o: Entidade C
displayName: Entidade C
```

Figura A.1: Pesquisa LDAP de utilizador com perfil administrador

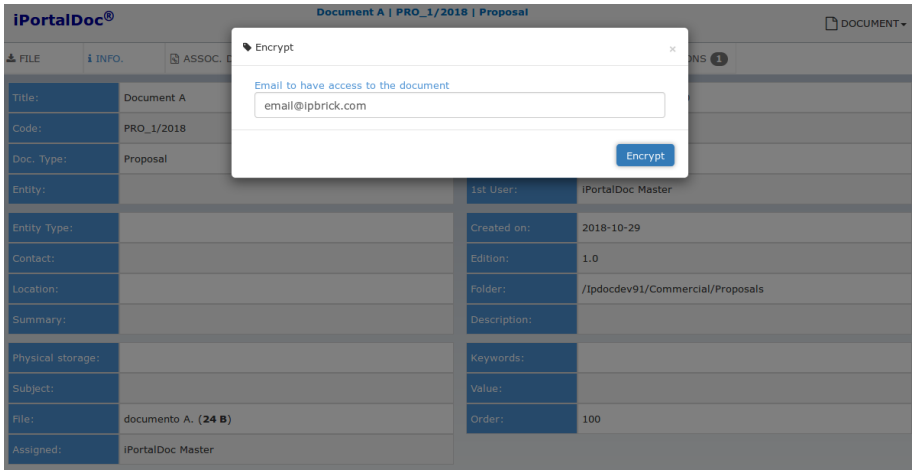

Figura A.2: Interface para encriptação

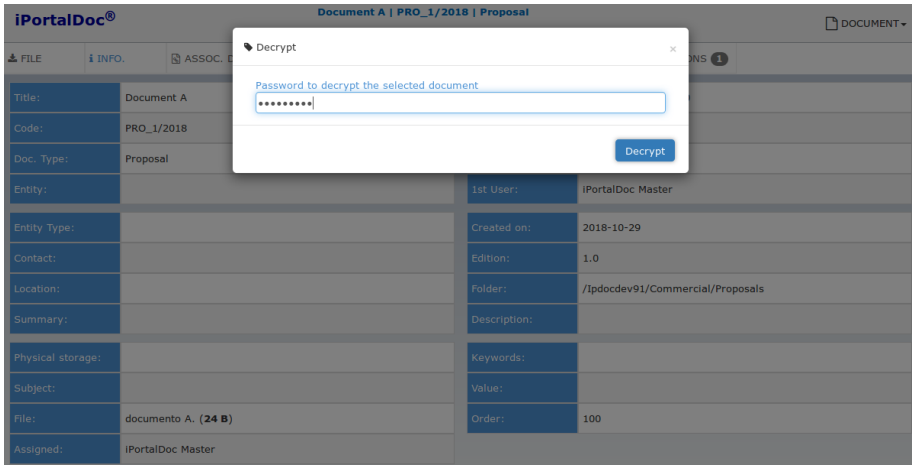

Figura A.3: Interface para desencriptação

Referências

[1] Aaron Woody. Enterprise Security: A Data-Centric Approach to Securing the Enterprise A guide to applying data-centric security concepts for securing enterprise data to enable an agile enterprise. 2013. Último acesso em 7 de Maio de 2018.

[2] GoAnywhere. OpenPGP Encryption Technology | GoAnywhere MFT, 2018. Último acesso em 14 de Maio de 2018. URL: https://www.goanywhere.com/managed-file-transfer/encryption/open-pgp.

[3] Oracle. Oracle Data Masking. Último acesso em 14 de Maio de 2018. URL: http://www.oracle.com/technetwork/database/options/data-masking-subsetting/overview/index.html.

[4] Informatica. Best Practices for Dynamic Data Masking Securing Production Applications and Databases in Real-Time. 2011. Último acesso em 15 de Maio de 2018. URL: https://www.informatica.com/downloads/1844_DDM_BestPractices_wp.pdf.

[5] Carlos José Pereira da Rocha. IPBrick - Contact Center para gestão de suporte a clientes. 7 2016. Último acesso em 20 de Outubro de 2018.

[6] Comodo. Comodo mydlp presentation, 2015. Último acesso em 22 de Dezembro de 2018. URL: https://www.slideshare.net/truongminhyen/comodo-my-dlptechpresentation060615v3.

[7] uniIT. UNI-IT - Data-Centric Security. Último acesso em 6 de Maio de 2018. URL: http://www.uni-it.com.br/data-security.

[8] OpenText Corp. GDPR: Start a plan for compliance | OpenText, 2018. Último acesso em 7 de Maio de 2018. URL: https://www.opentext.com/campaigns/infosec-compliance/gdpr?utm_source=youtube&utm_medium=ppc&utm_campaign=ent-ecm&utm_content=gdpr-aiim-information-privacy-and-security-ebook&elqcampaignid=30211.

[9] Gary Miglicco. GDPR is here and it is time to get serious. *Computer Fraud and Security*, 2018. Último acesso em 14 de Janeiro de 2019.

[10] Mike Bilger, Luke O 'connor, Matthias Schunter, Morton Swimmer, e Nev Zunic. Address changing security requirements in a dynamic business environment. 2006. Último acesso em 21 de Abril de 2018.

[11] Brian Desmond, Joe Richards, Robbie Allen, e Alistair G Lowe-Norris. Active Directory. Relatório técnico. Último acesso em 16 de Outubro de 2018.

[12] Yi Shiung Yeh, Wei Shen Lai, e Chung Jaye Cheng. Applying lightweight directory access protocol service on session certification authority. *Computer Networks*, 2002. Último acesso em 14 de Novembro de 2018.

[13] Charles Carrington, Timothy Speed, Juanita Ellis, Steffano Korper, Amsterdam Boston, London New, York Oxford, Paris San, Diego San, Francisco Singapore, e Sydney Tokyo. Enterprise Directory and Security Implementation Guide - Designing and Implementing Directories in your Organization. Relatório técnico, 2002. Último acesso em 16 de Outubro de 2018.

[14] Dan Raywood. GDPR - Companies Unprepared, Don't Know Where Data Is - Infosecurity Magazine, 2017. Último acesso em 7 de Maio de 2018. URL: https://www.infosecurity-magazine.com/news/gdpr-companies-unprepared-where/.

[15] Jennifer Bayuk. Data-centric security. *Computer Fraud and Security*, 2009. Último acesso em 21 de Abril de 2018.

[16] Matt Little. Data-Centric Security: Protecting What Really Matters - Infosecurity Magazine, 2017. Último acesso em 7 de Maio de 2018. URL: https://www.infosecurity-magazine.com/opinions/data-centric-security-protecting/.

[17] Accenture. ACHIEVING DATA-CENTRIC SECURITY HOW TO FEND OFF BREACHES BY BEING BRILLIANT AT THE BASICS. 2017. Último acesso em 21 de Abril de 2018.

[18] Ellen Zhang. What is Role-Based Access Control (RBAC)? Examples, Benefits, and More | Digital Guardian, 2017. Último acesso em 9 de Maio de 2018. URL: https://digitalguardian.com/blog/what-role-based-access-control-rbac-examples-benefits-and-more.

[19] Lior Arbel. Data loss prevention: The business case. *Computer Fraud and Security*, 2015. Último acesso em 14 de Novembro de 2018.

[20] Nate Lord. What is Network Data Loss Prevention? | Digital Guardian, 2015. Último acesso em 13 de Maio de 2018. URL: https://digitalguardian.com/blog/what-network-data-loss-prevention.

[21] Adrian Lane. Securosis - Blog - Article, 2014. Último acesso em 14 de Maio de 2018. URL: https://securosis.com/blog/trends-in-data-centric-security-tools.

[22] TokenEx. What is Tokenization? | Tokenization 101 | TokenEx, 2017. Último acesso em 14 de Maio de 2018. URL: https://tokenex.com/resource-center/what-is-tokenization/.

[23] Symantec. Symantec Data Loss Prevention At a glance Highest level of data protection Single pane of glass Wide range of integrations Keep data safe while in use on endpoints. 2017. Último acesso em 15 de Maio de 2018. URL: https://www.symantec.com/content/dam/symantec/docs/data-sheets/data-loss-prevention-family-en.pdf.

[24] Thales. Geração de Tokens com Mascaramento Dinâmico de Dados da Vormetric, 2017. Último acesso em 15 de Maio de 2018. URL: https://pt.thalesesecurity.com/products/data-tokenization-masking-and-transformation/tokenization-data-masking.

[25] Hesham S. Ahmad, Issa M. Bazlamit, e Maha D. Ayoush. Investigation of Document Management Systems in Small Size Construction Companies in Jordan. Em *Procedia Engineering*, 2017. Último acesso em 20 de Setembro de 2018.

[26] VIENNA Advantage. 8 Features every Document Management System - DMS - must have -, 2015. Último acesso em 4 de Dezembro de 2018. URL: http://viennaadvantage.com/blog/business-hacks/8-features-every-document-management-system-dms-must-have/.

[27] Refsnes Data. PHP 5 Introduction, 2018. Último acesso em 11 de Maio de 2018. URL: https://www.w3schools.com/php/php_intro.asp.

[28] IPBRICK SA. Sobre IPBRICK - IPBRICK, 2018. Último acesso em 10 de Maio de 2018. URL: https://www.ipbrick.com/pt-pt/sobre-ipbrick/.

[29] IPBRICK SA. IPBRICK OS - IPBRICK, 2018. Último acesso em 10 de Maio de 2018. URL: https://www.ipbrick.com/pt-pt/ipbrick-os/.

[30] IPBRICK SA. iPortalDoc, solução de Gestão Documental e de Processos, 2018. Último acesso em 10 de Maio de 2018. URL: https://www.ipbrick.com/pt-pt/iportaldoc-software-gestao-documental/.

[31] IPBRICK SA. Centro de Contactos - IPBRICK, 2018. Último acesso em 10 de Maio de 2018. URL: https://www.ipbrick.com/pt-pt/ipbrick-contactcenter/.