

UNIVERSIDADE ABERTA



Sistema de Servidor Pessoal de E-mail (Relatório Final – Versão Final)

**Ricardo Ferreira da Conceição Dias Marques
(Estudante nº 1100281)**

**“Projeto Final” (U.C. 21095)
da Licenciatura em Informática**

**Trabalho de Projeto orientado pelo
Professor Doutor Vitor Rocio**

4 de dezembro de 2016

ÍNDICE

| | |
|---|----|
| 1. Objetivos do Projeto e Resultados Esperados..... | 4 |
| 2. O Ambiente de Desenvolvimento e Testes..... | 5 |
| 2.1. Alojamento de máquina virtual remota na DigitalOcean..... | 5 |
| 2.2. Componentes de <i>software</i> do Ambiente de Desenvolvimento e Testes..... | 7 |
| 3. Descrição e Enquadramento do Projeto..... | 7 |
| 3.1. Funcionalidade adicional especificada pelo Professor Orientador..... | 8 |
| 3.1.1. Modelo de Dados da Solução para a Funcionalidade adicional..... | 10 |
| 3.1.1.1. Explicação do Modelo de Dados desenhado..... | 10 |
| 3.1.1.2. Comandos DDL associados ao Modelo de Dados da BD “dbmailings”..... | 15 |
| 4. Desenvolvimento do Projeto..... | 16 |
| 4.1. Registo do Domínio “uabdemo.info”..... | 16 |
| 4.2. Registos DNS do Domínio “uabdemo.info”..... | 17 |
| 4.3. Funcionalidade proposta pelo Professor Orientador..... | 22 |
| 4.3.1. Código Perl para processamento de mensagens e inserção na Base de Dados..... | 23 |
| 4.3.2. Código PHP para processamento e visualização da informação previamente guardada em Base de Dados..... | 25 |
| 5. Manual de Utilização..... | 25 |
| 5.1. Webmail – Roundcube..... | 25 |
| 5.2. Página web do Relatório de Mailings..... | 27 |
| 6. Calendarização do Projeto..... | 29 |
| 7. Conclusões..... | 30 |
| 7.1. Objetivos atingidos..... | 30 |
| 7.2. Oportunidades futuras de melhoria..... | 31 |
| 8. Software utilizado neste Relatório..... | 33 |
| 9. Referências Bibliográficas..... | 34 |
| 10. Anexos..... | 35 |
| 10.1. Ficheiro Perl “messages_metadata_parser_to_db.pl”..... | 35 |
| 10.2. Ficheiro PHP “mailings.php”..... | 45 |
| 10.3. Comandos executados na máquina virtual..... | 49 |
| 10.3.1. Instalação e configuração do Apache..... | 49 |
| 10.3.2. Instalação e configuração do MySQL..... | 50 |
| 10.3.3. Comandos para gerar o par de chave privada e pública RSA do certificado..... | 52 |
| 10.3.4. Comandos para gerar o CSR (Certificate Signing Request) do Certificado..... | 53 |
| 10.3.5. Instalação e configuração do Postfix..... | 54 |
| 10.3.6. Instalação e configuração do Dovecot..... | 56 |
| 10.3.7. Instalação e configuração do Roundcube..... | 62 |
| 10.3.8. Criação da Base de Dados de Mailings “dbmailings”..... | 64 |
| 10.3.9. Instalação de pacotes/módulos Perl..... | 65 |

Índice de Figuras

| | |
|--|----|
| Figura 1: Modelo de Dados da solução associado à funcionalidade adicional..... | 10 |
| Figura 2: Excerto da secção “3.6.4. Identification Fields” do RFC 5322..... | 14 |
| Figura 3: Código DDL do Modelo de Dados da funcionalidade adicional..... | 16 |
| Figura 4: Página da GoDaddy de Gestão de DNS para o domínio “uabdemo.info”..... | 17 |
| Figura 5: Página da DigitalOcean de configuração dos registo de DNS para o domínio “uabdemo.info”..... | 18 |
| Figura 6: Zone file do domínio “uabdemo.info”..... | 19 |
| Figura 7: Ligação ao servidor de IMAP, em Perl, usando o módulo IMAPClient..... | 24 |
| Figura 8: Código, em Perl, para fazer a ligação a uma Base de Dados MySQL..... | 24 |
| Figura 9: Código, em PHP, para fazer a ligação a uma Base de Dados MySQL..... | 25 |
| Figura 10: Página de login do Roundcube Webmail..... | 26 |
| Figura 11: Página da caixa de entrada (“Inbox”) do Roundcube Webmail..... | 27 |
| Figura 12: Screenshot da página Web do Relatório de Mailings..... | 28 |

Índice de Tabelas

| | |
|--|----|
| Tabela 1: Calendarização do Projeto..... | 29 |
|--|----|

O Estudante procurou desenvolver, para o seu Projeto Final, a Proposta de Estudante “**Sistema de Servidor Pessoal de E-mail**” tendo, como Orientador, o **Professor Doutor Vitor Rocio** (designado, neste Relatório, por “Professor Orientador”).

1. Objetivos do Projeto e Resultados Esperados

O objetivo principal do projeto consistiu em montar, num computador (servidor) com Sistema Operativo Linux, um sistema de e-mail bastante funcional, para o uso pessoal de um utilizador que queira administrar o seu próprio servidor de e-mail.

O sistema contempla múltiplas funcionalidades (que poderão ser entendidas como sendo os restantes objetivos do projeto), nomeadamente as seguintes:

- Envio de e-mail por SMTP com cifra do canal de transporte.
- Receção de e-mail via IMAPS (*Internet Message Access Protocol over SSL*), igualmente com cifra do canal de transporte.
- Suporte a múltiplos domínios de e-mail.
- Suporte a múltiplas *mailboxes* e *aliases* (permitindo reencaminhamento de mensagens para domínios internos ou externos), com informação das *mailboxes* e *aliases* registada em Base de Dados.
- *Interface web (Webmail)* para leitura e envio de e-mail, com certificado digital ativo para que as comunicações sejam feitas por HTTPS (*Hyper Text Transfer Protocol Secure*) para identificação do servidor e para cifrar o canal de transporte.
- Funcionalidade proposta pelo Professor Orientador: Criar uma funcionalidade de *reporting* automático no sistema de email que permita monitorizar entregas de documentos como resposta a um pedido. A ideia seria, partindo de um e-mail enviado a um grupo de pessoas, que o sistema permita criar um relatório em qualquer momento, que indique o número e percentagem de respostas recebidas,

com e sem ficheiros anexos. Esse relatório deveria também indicar quem do grupo de destinatários ainda não respondeu ao pedido (e eventualmente permitir a criação de um email de reforço apenas para esses).

2. O Ambiente de Desenvolvimento e Testes

2.1. Alojamento de máquina virtual remota na DigitalOcean

Para o desenvolvimento deste projeto, foi instalado e configurado, num computador com Sistema Operativo Linux, alojado na modalidade de VPS (*Virtual Private Server* – ou seja, numa máquina virtual alojada remotamente), todos os componentes de *software* necessários à implementação da solução do lado do servidor.

O fornecedor de VPS escolhido foi a DigitalOcean (<https://www.digitalocean.com/>) que é uma empresa norte-americana que fornece serviços de *cloud* do tipo IaaS (*Infrastructure-as-a-Service*). A DigitalOcean dá o nome de “*Droplets*” às máquinas virtuais que aloja. Para criar uma *droplet*, cada cliente da DigitalOcean escolhe¹:

- uma “Imagem” (“*Image*” na terminologia original) pré-criada e que pode ser:
 - ◊ uma de várias versões de algumas distribuições de Linux (nomeadamente CentOS, Fedora, Debian, Ubuntu, CoreOS) e FreeBSD (em várias versões disponíveis).
 - ◊ uma de várias “*One-click apps*” que consistem numa aplicação pré-instalada numa distribuição Linux, como seja, por exemplo: “*LAMP on 16.04*”, que significa um *stack* LAMP (Linux + Apache + MySQL + PHP) em Ubuntu 16.04. À data (novembro de 2016), as *One-Click Apps* disponíveis na DigitalOcean assentam todas na distribuição de Linux “Ubuntu” (versões 14.04 e 16.04), não estando disponíveis para outras distribuições de Linux ou FreeBSD.

¹ Informação obtida diretamente no *site* da DigitalOcean, observando as opções disponíveis na criação de uma *Droplet* e também em <http://searchcloudcomputing.techtarget.com/definition/DigitalOcean>

◊ *snapshots* ou *backups* que o cliente já tenha realizado de *Droplets* que tem ou que teve.

- Um tamanho / dimensionamento de *hardware* (“Size” na terminologia original) que é uma conjugação de número de processadores virtuais, memória RAM, espaço em disco (que é sempre do tipo SSD – *Solid State Disk* – o que obviamente traz benefícios de desempenho face aos tradicionais discos magnéticos rotativos) e um limite mensal de transferência de dados.
- Uma escolha de região do *datacenter* onde a *Droplet* ficará alojada (estando atualmente disponíveis, para escolha, 3 *datacenters* em Nova Iorque, 2 em São Francisco, 1 em Singapura, 1 em Londres, 1 em Frankfurt, 1 em Toronto e 1 em Bangalore).
- Opções adicionais, tais como a ativação de *backups*, a utilização de endereçamento privado de rede, IPv6. No caso de o cliente escolher *backups*, o método de *backup usado* é através do método de *snapshot*. A opção de *backups* tem um custo financeiro adicional de 20% face ao valor mensal do alojamento sem *backups*.

De acordo com a informação disponível na FAQ técnica da Digital Ocean (<https://www.digitalocean.com/help/technical/general/>), a tecnologia de virtualização (hipervisor) utilizada pela DigitalOcean é KVM (*Kernel-based Virtual Machine*), que é um *software (open-source)* de virtualização (<http://www.linux-kvm.org/>).

A configuração específica escolhida para a *Droplet* usada neste projeto foi a seguinte:

- Sistema Operativo: Distribuição de Linux CentOS, versão 6.8 (de 64 bits).
- O “tamanho” (*size*) escolhido foi o mais pequeno e barato disponível, correspondente à opção de 5 USD por mês, que dá direito a um processador virtual, 512 MB de Memória RAM, 20 GB de disco SSD e (até) 1000 GB de transferência de dados por mês.
- O *datacenter* escolhido foi o de Londres.

- Foi ainda selecionada a opção de *backups*. Ficou por implementar o backup via “mysqldump” para garantir que os backups são consistentes, no que diz respeito às Bases de Dados.

2.2. Componentes de *software* do Ambiente de Desenvolvimento e Testes

A solução usada assentou em *software* livre (*open source*). Mais especificamente, os componentes instalados e configurados foram os seguintes:

- *Software* servidor de Envio de E-mail (MTA – Mail Transfer Agent): Postfix (versão 2.6.6)
- *Software* servidor de Receção de E-mail: Dovecot (versão 2.0.9)
- *Software* de Webmail: Roundcube (versão 1.0.9)
- *Software* servidor HTTP (Hypertext Transfer Protocol) para suporte ao Webmail: Apache (versão 2.2.15)
- Linguagem de *scripting* Web para suporte ao Webmail e ao requisito adicional proposto pelo Professor Orientador: PHP (versão 5.3.3)
- *Software* servidor de Base de Dados (para suporte à Base de Dados de *mailboxes/aliases* e ao Webmail): MySQL (versão 5.1.73)

3. Descrição e Enquadramento do Projeto

Este projeto, conforme já foi descrito, assenta em montar um Sistema de Servidor de E-mail. No título deste projeto, é referido que é um servidor “**pessoal**” mas convém esclarecer que esse termo “pessoal” é usado no sentido de ter de existir uma pessoa que queira ser o Administrador do servidor de e-mail. Dito isso, como se irá ver, o sistema suporta múltiplas *mailboxes* e até múltiplos domínios de correio eletrónico, não existindo

assim qualquer limitação de o sistema apenas suportar um utilizador; pelo contrário, o sistema suporta múltiplos utilizadores.

O sistema implementado suporta o acesso ao correio eletrónico de duas formas diferentes:

- usando um *browser web* para aceder a um sistema de Webmail que, neste caso, é suportado no *software* chamado “*Roundcube*”. O acesso, para maior segurança, é feito de forma cifrada, usando HTTPS
- e através de um programa cliente de e-mail, em particular de um cliente que suporte o protocolo IMAP. Também para maior segurança, o acesso pode ser feito de forma cifrada, usando SSL/TLS. Foi usado o cliente *Mozilla Thunderbird* no desenvolvimento deste projeto, mas outros clientes de e-mail também deverão funcionar, como será o caso, por exemplo, do *Microsoft Outlook*, (*Apple*) *Mail* ou o *Evolution*.

3.1. Funcionalidade adicional especificada pelo Professor Orientador

Conforme foi referido anteriormente, o Professor Orientador propôs a seguinte funcionalidade adicional: funcionalidade de *reporting* automático no sistema de email que permita monitorizar entregas de documentos em resposta a um pedido. A ideia seria, partindo de um email enviado a um grupo de pessoas, que o sistema permitisse criar um relatório em qualquer momento, que indicasse o número e percentagem de respostas recebidas, com e sem ficheiros anexos. Esse relatório deveria também indicar quem do grupo de destinatários ainda não respondeu ao pedido (e eventualmente permitir a criação de um email de reforço apenas para esses).

Para a implementação desta funcionalidade, que foi parcialmente implementada, foi adotada a seguinte abordagem:

- Criação de uma *mailbox* de *mailings* para o controlo das mensagens que devem ser “controladas” no âmbito da funcionalidade adicional.
- Desenvolvimento de um programa/*script*, escrito na linguagem de programação Perl, que faça o *parsing* de meta-informação das mensagens de correio eletrónico em que um dos destinatários (que pode estar no campo “To:” ou “Cc:”) seja o endereço de e-mail associado à *mailbox* de *mailings* que foi indicado no ponto anterior. Esse programa alimenta uma Base de Dados relacional suportada no “motor” MySQL. O modelo de dados dessa Base de Dados irá ser apresentado e explicado na secção seguinte deste relatório.
- Desenvolvimento de uma página *web*, suportada na linguagem de programação PHP, que acedesse à Base de Dados indicada no ponto anterior e que, com base nos dados lidos, apresenta os relatórios correspondentes (em tabelas de HTML).

A opção pela utilização da linguagem de programação Perl, para o desenvolvimento do programa/*script* de *parsing da meta-informação das mensagens*, foi baseada:

- por um lado, nalgum conhecimento que o Estudante tem da linguagem Perl;
- por outro lado, na existência de vários módulos, no chamado CPAN (*Comprehensive Perl Archive Network*)² para o acesso programático a *mailboxes* (através do protocolo IMAP), de *parsing* de cabeçalhos de mensagens de correio eletrónico, etc...

² CPAN (*Comprehensive Perl Archive Network*) - <http://www.cpan.org/>

3.1.1. Modelo de Dados da Solução para a Funcionalidade adicional

Foi criada uma Base de Dados com o nome de “dbmailings” com o Modelo de Dados que se ilustra na seguinte imagem:

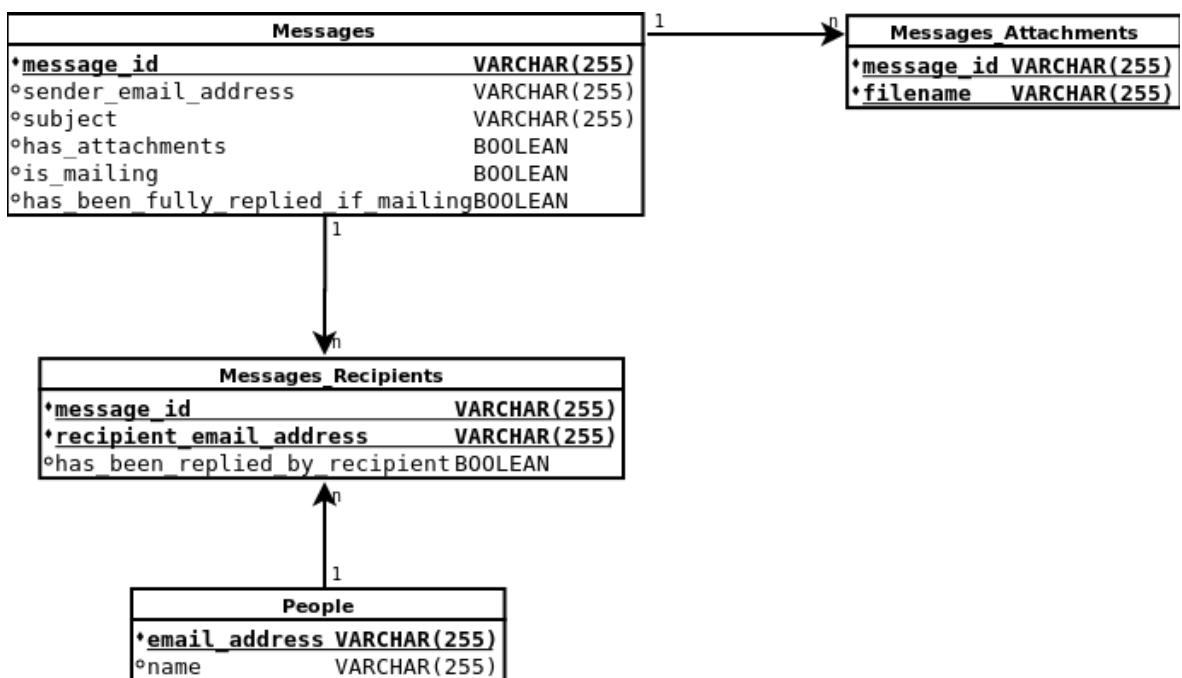


Figura 1: Modelo de Dados da solução associado à funcionalidade adicional

3.1.1.1. Explicação do Modelo de Dados desenhado

Conforme se pode ver acima, o Modelo de Dados consiste em 4 tabelas:

- Tabela de “**Messages**” com os seguintes campos (associados a cada mensagem):
 - ◊ **message_id**: chave primária da tabela. É um valor do tipo *string* e, como tal, foi usado o tipo de dados *VARCHAR* para o guardar. O valor deste campo é obtido através do valor do campo de “*Message-ID*” disponível em cada mensagem de

correio eletrónico. Mais adiante, será explicado o motivo da escolha deste campo como chave primária.

- ◊ `sender_email_address`: endereço de correio eletrónico do remetente da mensagem. Uma mensagem de correio eletrónico só terá um remetente (*sender*) que constará do campo “*From*”. O campo “*From*” poderá ter ainda (para além do endereço de correio eletrónico) o nome do remetente, mas é apenas o endereço de email que ficará guardado neste campo da tabela. Já no que diz respeito aos destinatários (*recipients*) da mensagem, poderá haver mais do que um e os mesmos campos “*To*” e/ou “*Cc*” (*Carbon Copy*) e/ou também do campo “*Bcc*” (*Blind Carbon Copy*) – embora, por simplificação, se tenha optado, neste projeto, por não considerar a possibilidade de utilização do campo de “*Bcc*”. Dado que poderá haver mais do que um destinatário de uma mensagem, a identificação dos (endereços de email dos) destinatários foi movida para uma outra tabela (à qual foi dado o nome de “**Messages_Recipients**”).
- ◊ `subject`: corresponde ao campo com o mesmo nome (“*Subject*”) nas mensagens de correio eletrónico, onde consta o “assunto” / “título” que o remetente dá à mensagem. Foi assumido, por simplificação, o pressuposto que uma mensagem que tenha um título “novo” (isto é, um título que **não** comece com um indicador/prefixo de resposta - “*Re*” ou “*RE*”) constitua um novo *mailing*; enquanto uma mensagem cujo assunto se inicie com um desses indicadores de resposta é considerado como sendo uma resposta ao *mailing* originado pela mensagem que tenha o mesmo título excluindo o indicador de resposta. Naturalmente, é possível que se dê o caso de haver uma mensagem com um assunto que tenha indicador de resposta para a qual não exista uma mensagem “original” correspondente mas, também por simplificação, não foi contemplada essa possibilidade no sistema. Um outro aspeto importante é que, independentemente da codificação de carateres que seja usada na mensagem (e, em particular, no assunto da mensagem), na Base de Dados irá ser sempre usada

a codificação de carateres Unicode UTF-8, de forma a tratar, de uma forma consistente, os carateres internacionais (como sejam os carateres portugueses acentuados).

- ◊ **has_attachments**: este é um campo do tipo booleano que simplesmente indica se existem ou não ficheiros anexados (também chamados de “anexos” ou *attachments*) à mensagem.
 - ◊ **is_mailing**: este é outro campo do tipo booleano que indica se a mensagem é ou não uma mensagem que inicia um *mailing*. Conforme foi referido na explicação do campo **subject**, é assumido que uma mensagem cujo assunto **não** se inicie com um indicador/prefixo de resposta (“*Re:*” ou “*RE:*”) é um *mailing* e essas mensagens terão o valor de “verdadeiro” - 1 ou “TRUE” - neste campo de **is_mailing**; enquanto mensagens cujo assunto se inicie com um indicador de resposta, terão este campo com valor de “falso” - 0 ou “FALSE”.
 - ◊ **has Been Fully Replied If Mailing**: este é um campo booleano de controlo que verifica se um *mailing* foi (ou não) respondido por todos os seus destinatários. A forma de concluir se um *mailing* foi ou não totalmente respondido é, para as mensagens que são de *mailing*, verificar se existe uma mensagem correspondente com o mesmo assunto, mas que se inicie com um indicador de resposta, por parte de cada um dos destinatários da mensagem de *mailing*. Numa outra tabela, a tabela “**Messages_Recipients**” será mantido o registo, para cada par de mensagem e de destinatário, se o destinatário já respondeu ou não.
- Tabela de “**Messages_Attachments**”, onde são registados os ficheiros anexados às mensagens. A chave primária desta tabela é uma chave primária composta, formada pelos dois campos que são os seguintes:
 - ◊ **message_id**: é um valor do tipo *string* sendo uma chave estrangeira (*Foreign Key*) que está associada ao campo **message_id** da tabela “**Messages**”.

- ◊ **filename**: contém o nome do ficheiro anexo. Sendo um valor do tipo *string*, foi usado o tipo de dados VARCHAR para o guardar.
- Tabela de “**People**”, onde é registada informação muito simples das pessoas que sejam remetentes e/ou destinatários de mensagens. Os campos desta tabela são os seguintes:
 - ◊ **email_address**: endereço de correio eletrónico da pessoa. Sendo um valor do tipo *string*, foi usado o tipo de dados VARCHAR para o guardar. Este campo foi definido como sendo a chave primária desta tabela o que, naturalmente, tem o pressuposto associado de cada pessoa registada no sistema ter/usar um e apenas um endereço de email. Este pressuposto foi assumido, pensando que a funcionalidade adicional tem em vista um cenário em que o remetente e os destinatários trabalhem todos numa mesma organização (por exemplo, numa empresa ou numa Universidade) e em que as pessoas dessa organização tenham um endereço de e-mail associado a um domínio dessa organização.
 - ◊ **name**: é o nome da pessoa. Sendo um valor do tipo *string*, foi usado o tipo de dados VARCHAR para o guardar. O nome do campo foi definido como opcional porque é possível que as mensagens que uma pessoa receba apenas mencionem o endereço de email e não o nome do destinatário.

Passamos então agora a explicar o motivo para a escolha do campo “**message_id**” como chave primária da tabela “**Messages**”:

Dado que o sistema deste Projeto assenta em mensagens de correio eletrónico, têm particular relevância as normas / *standards* internacionais que definem o formato dessas mensagens. As normas mais relevantes para o formato de mensagens são alguns RFC (*Request For Comments*) da comunidade internacional IETF (*Internet Engineering Task Force*), em particular o RFC 5322 – *Internet Message Format* (Resnick, 2008) que foi publicado em 2008 e que teve, como Editor, Pete Resnick da empresa QUALCOMM. Esse

RFC substituiu/tornou obsoleto o RFC 2822 (Resnick, 2001) que tinha sido publicado em 2001 (igualmente editado por Pete Resnick) e que, por sua vez, tinha tornado obsoleto o RFC 822 – *Standard for the format of ARPA Internet Text Messages* (Crocker, 1982).

Um dos campos que tem particular importância é o de “Message-ID” (que é apresentado como “message_id” no Modelo de Dados desenhado). Tal como é indicado no RFC 5322, em particular na secção “3.6.4. Identification fields”, esse campo de “Message-ID”, embora seja opcional, é recomendado (o que é denotado pela utilização da palavra “SHOULD”) e, caso exista, esse identificador da mensagem terá de ser único:

3.6.4. Identification Fields

Though listed as optional in the table in [section 3.6](#), every message SHOULD have a "Message-ID:" field. Furthermore, reply messages SHOULD have "In-Reply-To:" and "References:" fields as appropriate and as described below.

(...)

The "Message-ID:" field contains a single unique message identifier.

Figura 2: Excerto da secção “3.6.4. Identification Fields” do RFC 5322

O significado da palavra “SHOULD”, neste contexto, é explicado no RFC 2119 – *Key words for use in RFCs to Indicate Requirement Levels*. (Bradner, 1997): “*This word, or the adjective "RECOMMENDED", mean that there may exist valid reasons in particular circumstances to ignore a particular item, but the full implications must be understood and carefully weighed before choosing a different course.*”.

3.1.1.2. Comandos DDL associados ao Modelo de Dados da BD “dbmailings”

Para referência, aqui se incluem os comandos DDL (*Data Definition Language*) que foram usados para criar as tabelas, campos, chaves primárias, chaves estrangeiras e índices do Modelo de Dados da Base de Dados “dbmailings”:

```
CREATE TABLE IF NOT EXISTS `Messages` (
  `message_id` VARCHAR(255),
  `sender_email_address` VARCHAR(255),
  `subject` VARCHAR(255),
  `has_attachments` BOOLEAN DEFAULT 0,
  `is_mailing` BOOLEAN DEFAULT 0,
  `has_been_fully_replied_if_mailing` BOOLEAN DEFAULT 0,
  PRIMARY KEY (`message_id`)
)
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8
COLLATE = utf8_bin;

CREATE TABLE IF NOT EXISTS `Messages_Attachments` (
  `message_id` VARCHAR(255),
  `filename` VARCHAR(255),
  INDEX `message_id_ind`(`message_id`),
  PRIMARY KEY (`message_id`, `filename`),
  FOREIGN KEY (`message_id`) REFERENCES `Messages`(`message_id`)
)
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8
COLLATE = utf8_bin;

CREATE TABLE IF NOT EXISTS `People` (
  `email_address` VARCHAR(255),
  `name` VARCHAR(255),
  PRIMARY KEY (`email_address`)
)
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8
COLLATE = utf8_bin;

CREATE TABLE IF NOT EXISTS `Messages_Recipients` (
  `message_id` VARCHAR(255),
  `recipient_email_address` VARCHAR(255),
  `has_been_replied_by_recipient` BOOLEAN DEFAULT 0,
  INDEX `message_id_ind`(`message_id`),
  INDEX `recipient_email_address_ind`(`recipient_email_address`),
```

```
PRIMARY KEY (`message_id`, `recipient_email_address`),  
FOREIGN KEY (`message_id`) REFERENCES `Messages`(`message_id`),  
FOREIGN KEY (`recipient_email_address`) REFERENCES  
`People`(`email_address`)  
)  
ENGINE = InnoDB  
DEFAULT CHARACTER SET = utf8  
COLLATE = utf8_bin;
```

Figura 3: Código DDL do Modelo de Dados da funcionalidade adicional

Conforme se vê, a partir da leitura do código DDL acima, é sempre usada a codificação de caracteres Unicode UTF-8, para lidar com os caracteres internacionais.

4. Desenvolvimento do Projeto

4.1. Registo do Domínio “uabdemo.info”

Como já foi referido, o sistema criado suporta múltiplos domínios (de e-mail). Para este Projeto foi registado um domínio de DNS chamado “uabdemo.info”. O *registrar* escolhido para registrar o domínio foi a GoDaddy.com

Na configuração do domínio na GoDaddy, foram colocados os *nameservers* da DigitalOcean de forma a poder fazer a criação dos vários registos de DNS do domínio “uabdemo.info” usando, para tal o painel de controlo de DNS da DigitalOcean.

Abaixo ilustra-se a página de “**Gestão de DNS**” da GoDaddy, para o domínio “uabdemo.info”, onde se pode ver que foram colocados os “**Servidores de nome**” (*nameservers*) da DigitalOcean (ns1.digitalocean.com, ns2.digitalocean.com e ns3.digitalocean.com):

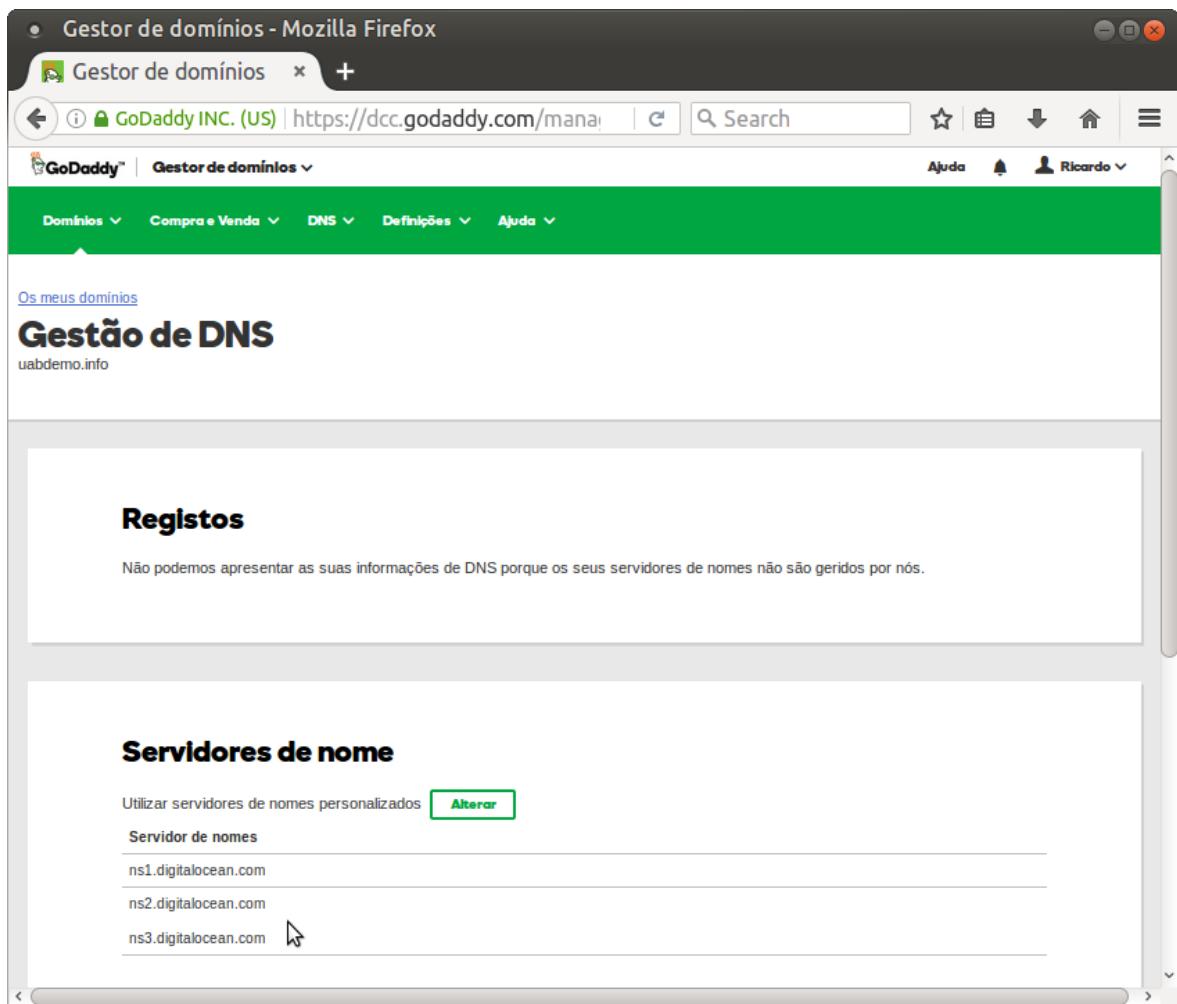


Figura 4: Página da GoDaddy de Gestão de DNS para o domínio “uabdemo.info”

4.2. Registros DNS do Domínio “uabdemo.info”

Passa agora a ilustrar-se a configuração dos registos de DNS que foi realizada, no painel de controlo da DigitalOcean, para o domínio “uabdemo.info” - gerido via URL <https://cloud.digitalocean.com/domains/uabdemo.info> :

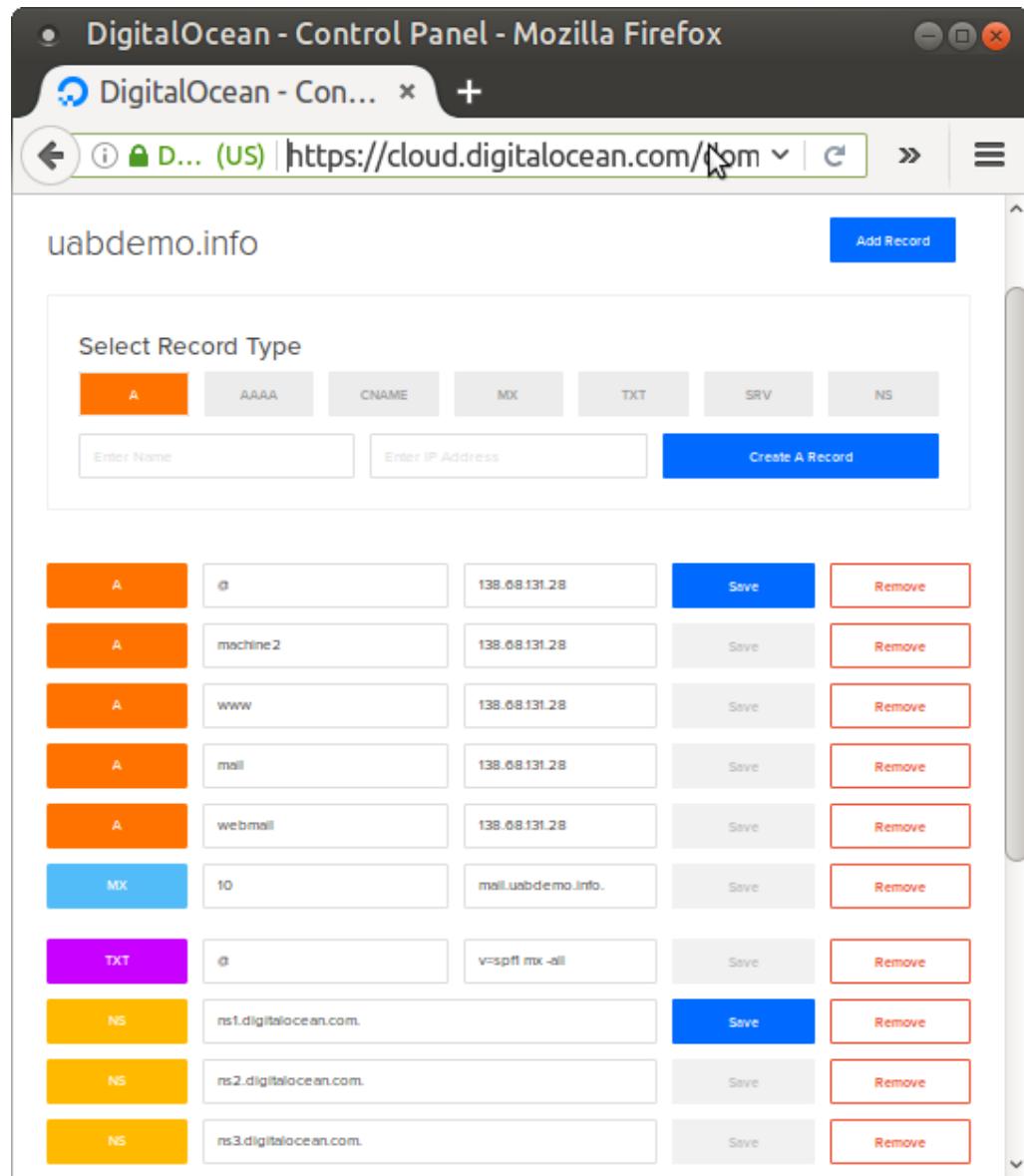


Figura 5: Página da DigitalOcean de configuração dos registo de DNS para o domínio “uabdemo.info”

Como é explicado em Coulouris (2011), os dados do sistema de nomes do DNS estão divididos por zonas em que uma zona contém a seguinte informação:

- os dados dos atributos para os nomes num domínios (excluindo algum subdomínio que seja administrado por autoridades de nível inferior).
- os nomes e endereços IP de pelo menos 2 servidores de nomes que forneçam dados autoritativos³ (*authoritative* na terminologia original inglesa) para a zona.
- os nomes dos servidores de nomes que contenham dados autoritativos para subdomínios que sejam delegados.
- Parâmetros de gestão de zona, como por exemplo dados relativos a *caching* e a replicação.

No caso do domínio “uabdemo.info” que foi criado para este projeto, os registo de DNS deste domínio “uabdemo.info” (constantes da *screenshot* do painel de controlo de DNS da DigitalOcean) traduzir-se-iam no seguinte “*Zone File*” de DNS (que também consta da página acima):

```
$ORIGIN uabdemo.info.
$TTL 1800
uabdemo.info. IN SOA ns1.digitalocean.com. hostmaster.uabdemo.info.
1477782678 10800 3600 604800 1800
uabdemo.info. 1800 IN NS ns1.digitalocean.com.
uabdemo.info. 1800 IN NS ns2.digitalocean.com.
uabdemo.info. 1800 IN NS ns3.digitalocean.com.
uabdemo.info. 1800 IN A 138.68.131.28
machine2.uabdemo.info. 1800 IN A 138.68.131.28
www.uabdemo.info. 1800 IN A 138.68.131.28
mail.uabdemo.info. 1800 IN A 138.68.131.28
uabdemo.info. 1800 IN MX 10 mail.uabdemo.info.
webmail.uabdemo.info. 1800 IN A 138.68.131.28
uabdemo.info. 1800 IN TXT v=spf1 mx -all
```

Figura 6: Zone file do domínio “uabdemo.info”

O formato de uma *zone file* encontra-se especificado no RFC 1034 – *Domain Names – Concepts and Facilities* (Mockapetris, 1987a) e no RFC 1035 – *Domain Names –*

³ Foi usado aqui o termo “autoritativo” como tradução do termo original “authoritative” por ser essa a tradução adotada no documento “Regulamento de Registo de Domínios .PT” - <https://www.dns.pt/pt/dominios-2/regras-de-dominios/capitulo-i/>

Implementation and Specification (Mockapetris, 1987b), em particular na secção “5. MASTER FILES” do RFC 1034.

Várias das linhas da *zone file* dizem respeito ao que é identificado no RFC 1034 como sendo RR (*Resource Records*). Nesse RFC, é identificado que a estrutura de uma linha de RR é a seguinte:

owner type class TTL

Em que:

- *owner* é o nome de domínio do RR
- *type* é o tipo de recurso (podendo ser “A”, “CNAME”, “HINFO”, “MX”, “NS”, “PTR” e “SOA”. O RFC 1035 ainda acrescenta alguns outros tipos, ganhando relevância, neste contexto, o tipo “TXT”)
- *class* refere-se a uma família de protocolos, podendo ter o valor de “IN” (relativo ao “Internet System”) ou de “CH” (que, por estranho que possa parecer, se refere a um sistema chamado “Chaos System”, relativa a uma rede local originada pelo MIT⁴). Em quase todos os casos, o valor que aparece é “IN”.
- TTL significa *Time-To-Live* e representa o tempo após o qual um cliente de DNS deverá descartar a informação de resolução de nome que possa ter mantido em *cache* e refrescar a informação a partir da fonte original.

No caso dos registos de tipo de recurso, ganham particular relevância os seguintes tipos:

- “A” - que associa o nome de um *host* em particular a um endereço IP
- “NS” - que associa um domínio a um *nameserver*
- “MX” - que indica um *host* responsável por ser um *Mail Exchanger*, ou seja um servidor responsável por receber o e-mail que seja dirigido a um domínio. A linha

⁴ Para mais informação sobre a Chaosnet, consultar, por exemplo, o documento de 1981 disponível em <https://dspace.mit.edu/handle/1721.1/6353>

correspondente na “*zone file*” também tem um valor numérico que indica a prioridade de cada um dos servidores de e-mail, caso constem vários (um valor mais baixo indica um servidor mais “preferível”).

- “*TXT*” - refere-se a um *Text Record* e serve para inserir algum texto. Um dos usos deste campo é o de guardar informação de SPF (*Sender Policy Framework*) que foi o uso dado neste domínio “uabdemo.info” e que irá ser explicado um pouco mais à frente.

No caso do domínio/zona “uabdemo.info”, temos então várias linhas no *zone file* que indicam que o mesmo tem a responsabilidade pela resolução dos nomes assegurada pelos name servers. Assim, as 3 linhas seguintes, relativas ao tipo “*NS*” (*Nameserver*) indicam que a zona “uabdemo.info” tem a resolução dos nomes assegurada pelos *nameservers* “ns1.digitalocean.com”, “ns2. digitalocean.com” e “ns3.digitalocean.com” e que o TTL (*Time-To-Live*) correspondente é de 1800 segundos (30 minutos):

```
uabdemo.info. 1800 IN NS ns1.digitalocean.com.  
uabdemo.info. 1800 IN NS ns2.digitalocean.com.  
uabdemo.info. 1800 IN NS ns3.digitalocean.com.
```

Existem várias linhas dos chamados “*A Records*” que associam nomes de *hosts* aos endereços IP respetivos. No caso do domínio criado (“uabdemo.info”) - e como se pode ver, pela leitura das linhas correspondentes do *zone file* - o endereço IP usado nesta zona é sempre o 138.68.131.28 que corresponde ao endereço público IPv4 associado à *Droplet* criada (e também usando um TTL de 1800 segundos):

```
machine2.uabdemo.info. 1800 IN A 138.68.131.28  
www.uabdemo.info. 1800 IN A 138.68.131.28  
mail.uabdemo.info. 1800 IN A 138.68.131.28  
webmail.uabdemo.info. 1800 IN A 138.68.131.28
```

Sendo este um projeto de montar um servidor de e-mail, naturalmente teria de ser definido um (ou mais) registo(s) MX (*Mail Exchanger*). Neste caso, foi definido que o host *mail.uabdemo.info* iria ser o *host* responsável por essa função, com uma prioridade de 10 (embora, como não existem outros servidores de e-mail, a prioridade não seja relevante):

uabdemo.info. 1800 IN MX 10 mail.uabdemo.info.

A última linha da *zone file* do domínio “uabdemo.info” diz respeito a um registo DNS do tipo “TXT”. Tal como foi referido anteriormente, os registo “TXT” (*Text Record*) guardam informação textual.

No caso do domínio “uabdemo.info”, o único registo “TXT” serve para guardar informação relativa a SPF (*Sender Policy Framework*). O SPF encontra-se especificado no RFC 7208 – *Sender Policy Framework (SPF) for Authorizing Use of Domains in Email, Version 1* (Kitterman, 2014) que substituiu o RFC 4408 com o mesmo título (Wong & Schlitt, 2006). Conforme é explicado nesses RFC, o SPF visa dificultar a falsificação da origem de um e-mail e funciona explicitando quais são os *hosts* (servidores) que estão autorizados a enviar e-mail usando os nomes de domínio que constem do registo DNS correspondente. Para o domínio “uabdemo.info”, a linha relevante é a seguinte:

uabdemo.info. 1800 IN TXT v=spf1 mx -all

Conforme é explicado em http://www.openspf.org/SPF_Record_Syntax a expressão “mx” indica que o envio de e-mail é permitido aos servidores que tenham sido definidos como *Mail Exchangers* (isto é, que tenham um registo do tipo MX associado), sendo proibido a todos os outros servidores (o que é indicado pela expressão “-all”).

4.3. Funcionalidade proposta pelo Professor Orientador

Para implementar a funcionalidade proposta pelo Professor Orientador, foi adotada a abordagem de criar um programa/script na linguagem de programação Perl que executa o processamento/parsing das mensagens presentes na pasta “INBOX” de uma *mailbox* de *mailings* (associada ao endereço mailings@uabdemo.info) e que guarda, numa base de dados MySQL, a meta-informação processada.

Para mostrar a meta-informação processada ao utilizador, foi criada um programa/página *web* na linguagem de programação PHP, que lê a informação da base de

dados MySQL e a mostra ao utilizador. A página pode ser consultada no URL <http://www.uabdemo.info/mailings.php>

Uma implicação desta abordagem é que é necessário executar o programa Perl sempre que se queira “refrescar” a informação que está guardada em Base de Dados. Uma possibilidade seria agendar a execução do programa Perl em *crontab* para o mesmo ser executado, de forma automática, com bastante frequência. Uma alternativa, provavelmente ainda mais eficiente, seria a de estudar alguma solução para que, sempre que o servidor de e-mail (Dovecot) recebesse uma nova mensagem na *mailbox* de *mailings*, fosse executado então (e só nesses momentos) o referido programa Perl.

Uma outra melhoria que não foi implementada, por falta de tempo, seria a de implementar lógica, no programa Perl, para mover, para uma outra pasta da *mailbox de mailings*, as mensagens que já tivessem sido guardadas numa Base de Dados MySQL, de forma a reduzir o volume de mensagens que o programa tem de processar em cada execução. Nesse cenário alternativo, só teriam de ser processadas, em cada execução, as mensagens cuja meta-information não tivesse já sido guardada na Base de Dados.

4.3.1. Código Perl para processamento de mensagens e inserção na Base de Dados

O programa Perl criado acede, de forma programática, usando o protocolo IMAP, à *mailbox* de *mailings*.

Neste programa, foram usados vários módulos, do chamado CPAN (*Comprehensive Perl Archive Network*)⁵. Um dos módulos usados foi o “**IMAPClient**”, na sua versão 3.37⁶. A escolha da versão deveu-se a ser essa a versão que está disponível, nos repositórios de pacotes do CentOS, para o CentOS 6.8. Como o próprio nome indica, o módulo

⁵ CPAN (*Comprehensive Perl Archive Network*) - <http://www.cpan.org/>

⁶ Documentação da versão 3.37 do módulo IMAPClient: <http://search.cpan.org/~plobbes/Mail-IMAPClient-3.37/lib/Mail/IMAPClient.pod>

“IMAPClient” funciona como um cliente IMAP para fazer o acesso programático a *mailboxes*. Como o script Perl está instalado no próprio servidor de e-mail, o servidor usado é **localhost**. O código seguinte faz a ligação ao servidor de IMAP (apenas foi omitido, no excerto abaixo, a *password* original):

```
use Mail::IMAPClient;

use constant MAILINGS_EMAIL_ADDRESS => qw(mailings@uabdemo.info);

my $imap = Mail::IMAPClient->new(
    Server    => 'localhost',
    User      => MAILINGS_EMAIL_ADDRESS,
    Password  => '[Password do utilizador]',
    Ssl       => 0,
    Uid       => 0,
);
```

Figura 7: Ligação ao servidor de IMAP, em Perl, usando o módulo IMAPClient

Como se vê pelo código acima, a ligação é feita usando o nome do servidor, o *username* e a *password* da mailbox e a indicação se é ou não usada cifra do canal de transporte via SSL (dado que o acesso é feito no próprio servidor, não se justificará a cifra do canal de transporte).

Dado que a informação é guardada numa Base de Dados MySQL, foi também usado um módulo Perl do CPAN para fazer as operações relacionadas com a Base de Dados. Neste caso, foi usado o módulo DBI (*Database Interface*), na sua versão 1.609, para fazer essas operações. O código relevante para fazer a ligação à Base de Dados é o seguinte (apenas foi omitido, no excerto abaixo, a *password* original):

```
# Connect to MySQL Database
my $db_username = 'mailings_dbuser';
my $db_password = '[Password do utilizador]';
my $db_server = 'localhost';
my $db_name = 'dbmailings';

my $dbh = DBI->connect("DBI:mysql:$db_name:$db_server", $db_username,
$db_password)
    or die("ERROR connecting to the Database");
```

Figura 8: Código, em Perl, para fazer a ligação a uma Base de Dados MySQL

4.3.2. Código PHP para processamento e visualização da informação previamente guardada em Base de Dados

Tal como foi indicado anteriormente, foi criado um programa/página *web* na linguagem de programação PHP, que lê a informação da base de dados MySQL e a mostra ao utilizador. A página pode ser consultada no URL <http://www.uabdemo.info/mailings.php>

Para fazer a ligação à base de dados MySQL, foi usada a biblioteca/extensão MySQLi (*MySQL Improved Extension*).⁷ O código seguinte faz a ligação à Base de Dados MySQL (apenas foi omitido, no excerto abaixo, a *password* original):

```
$db_username = 'mailings_dbuser';
$db_password = '[Password do utilizador]';
$db_server = 'localhost';
$db_name = 'dbmailings';

$mysqli = new mysqli("$db_server", "$db_username", "$db_password",
"$db_name");
if ($mysqli->connect_errno) {
    echo "Failed to connect to MySQL: " . $mysqli->connect_error;
}
```

Figura 9: Código, em PHP, para fazer a ligação a uma Base de Dados MySQL

5. Manual de Utilização

5.1. Webmail – Roundcube

Na vertente de utilização do sistema de e-mail, os utilizadores que tenham um endereço de e-mail e a respetiva *mailbox* associada podem usar o sistema de e-mail através de um programa cliente de e-mail (como o *Mozilla Thunderbird*, o *Microsoft Outlook* ou outro) ou através do *Webmail*. Como já foi referido anteriormente, o sistema de *webmail* que foi instalado e configurado é o **Roundcube**.

⁷ Documentação sobre a biblioteca/extensão MySQLi disponível em <http://php.net/manual/en/book.mysql.php>

A página *web* do sistema de *Webmail* está localizada no seguinte URL:
<https://webmail.uabdemo.info/>

Abaixo, inclui-se uma *screenshot* da página de *login* no Webmail:

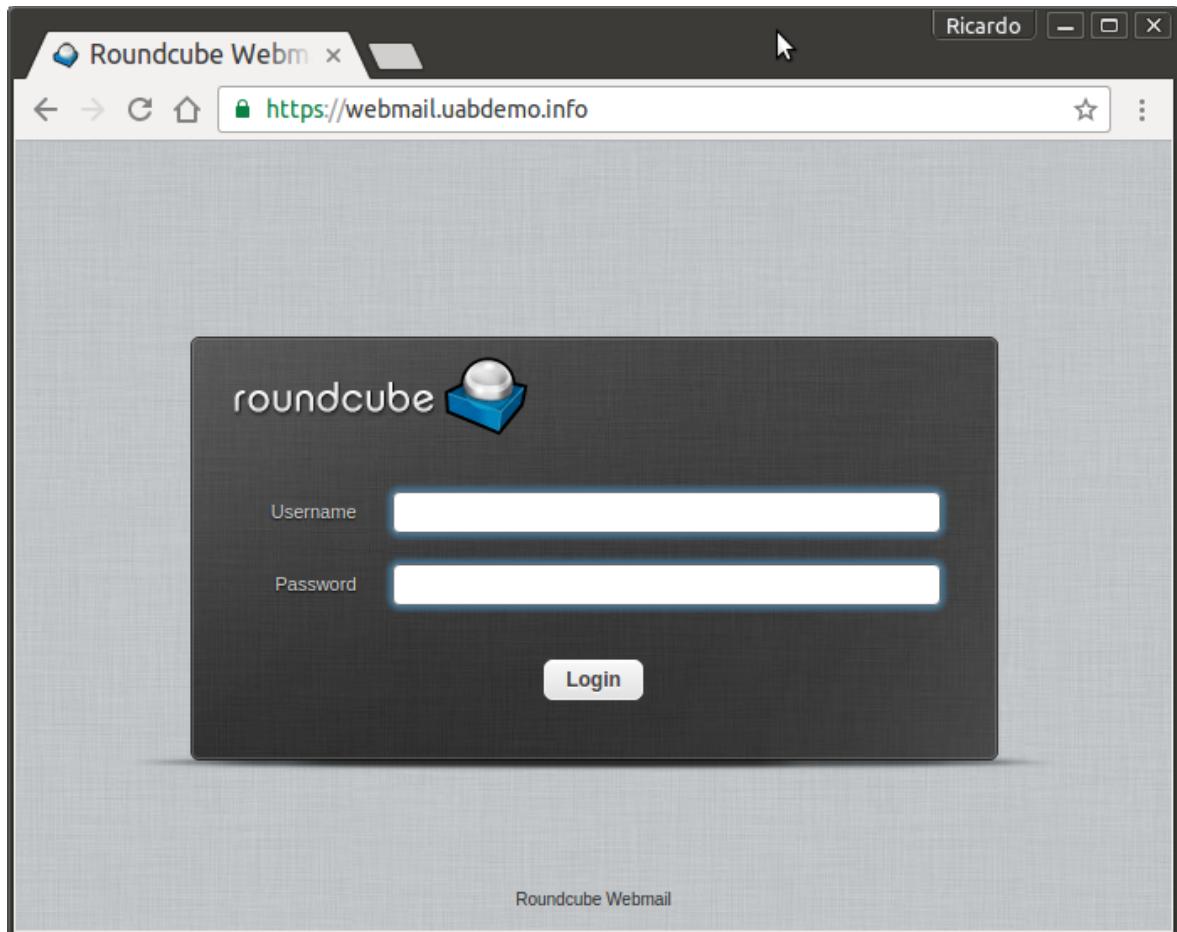


Figura 10: Página de *login* do Roundcube Webmail

O utilizador deverá fazer *login* com o seu endereço de e-mail na caixa de texto “**Username**” e introduzir a *password* (que lhe tenha sido transmitida pelo Administrador do sistema) na caixa de texto “**Password**”.

Após o *login* no *Webmail* ter sido concluído com sucesso, o utilizador acede à página correspondente à sua caixa de entrada (“**Inbox**”):

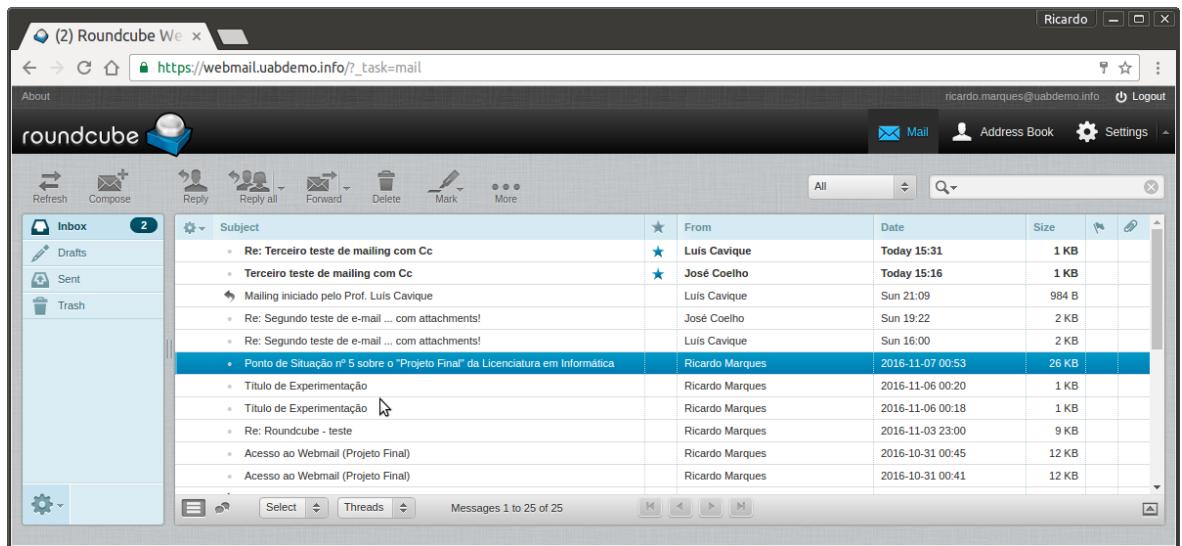


Figura 11: Página da caixa de entrada (“Inbox”) do Roundcube Webmail

Para mais informação sobre a utilização do *Roundcube Webmail*, remete-se o leitor para o Manual do próprio Roundcube: *Roundcube Webmail User Guide for Version 1.0* (https://docs.roundcube.net/doc/help/1.0/en_US/).

5.2. Página web do Relatório de Mailings

A página web de Relatório de *Mailings* (*Mailings Report*) está localizada no seguinte URL: <https://www.uabdemo.info/mailings.php>

Abaixo, inclui-se uma *screenshot* dessa página:

| Name | Email address |
|-----------------|------------------------------|
| José Coelho | jose.coelho@uabdemo.info |
| Luis Cavique | luis.cavique@uabdemo.info |
| Ricardo Marques | ricardo.marques@uabdemo.info |

Mailings

- Mailing #1

Mailing initiator: José Coelho
Subject: Terceiro teste de mailing com Cc
Message ID: c49c3d7b-05ad-6732-c07f-c80b278adfb3@uabdemo.info
Replied by all recipients: No

Recipients:

Name: Luis Cavique
Has replied: Yes
Has attachments: No

Name: Ricardo Marques
Has replied: No

- Mailing #2

Figura 12: Screenshot da página Web do Relatório de Mailings

Como se pode ver pela *screenshot* acima, a página inclui, no seu topo, uma lista de pessoas (“**People**”) - com o nome e endereço de e-mail de cada um - que tenham sido remetentes e/ou destinatários nalgum dos *mailings*. Logo abaixo dessa secção de “**People**”, aparece uma secção de “**Mailings**” (em que os vários *mailings* aparecem listados sem qualquer ordenação especial) e em que é indicado, para cada um dos *mailings*:

- quem o iniciou (“**Mailing initiator**”)
- o assunto do *mailing* (“**Subject**”)
- o *Message-ID* da mensagem que deu início ao *mailing* (“**Message-ID**”)
- a indicação se o *mailing* foi (“**Yes**”) ou não (“**No**”) respondido por todos os seus destinatários (“**Replied by all recipients**”)
- para cada *mailing* aparece uma secção de destinatários (“**Recipients**”) em que, para cada destinatário, é indicado:
 - ◊ o nome do destinatário (“**Name**”)

- ◊ a indicação se o destinatário respondeu (“**Yes**”) ou não (“**No**”) ao *mailing* (“**Has replied**”)
- ◊ a indicação se a resposta, por parte do destinatário, tem (“**Yes**”) ou não (“**No**”) ficheiros anexos (“**Has attachments**”)
- ◊ no caso de a resposta, por parte do destinatário, ter ficheiros anexos, é indicado o nome de cada um dos ficheiros anexos (em linhas começadas por “**Attachment # _ in Reply**”),

6. Calendarização do Projeto

Para referência, aqui se inclui a Calendarização que esteve associada a este Projeto:

| Nome da Atividade | Nome da Tarefa | Duração (dias) | Início | Fim |
|--------------------------|---|----------------|------------------------------|------------------------------|
| Proposta Inicial | | | | |
| | Elaboração da Proposta Inicial | 9 | 11-09-2016 | 19-09-2016 |
| | Submissão da Proposta Inicial | 1 | 20-09-2016 | 20-09-2016 |
| Desenvolvimento e Testes | | | | |
| | Criação do Ambiente Local de Desenvolvimento | 11 | 27-09-2016 | 07-10-2016 |
| | Criação do Ambiente Remoto de Testes / demonstração | 20 | 07-10-2016 | 26-10-2016 |
| Relatório Intermédio | | | | |
| | Elaboração do Relatório Intermédio | 13 | 21-10-2016 | 02-11-2016 |
| | Entrega de Relatório Intermédio para Aceitação | 1 | 03-11-2016 | 03-11-2016 |
| | Submissão do Relatório Intermédio | 1 | 07-11-2016 | 07-11-2016 |
| Relatório Final | | | | |
| | Elaboração do Relatório Final | 17 | 09-11-2016 | 25-11-2016 |
| | Entrega do Relatório Final para Aceitação | 1 | 25-11-2016 | 25-11-2016 |
| | Submissão do Relatório Final | 1 | 30-11-2016 | 30-11-2016 |
| Defesa | | | | |
| | Defesa do Projeto Final | 1 | Dez-2016 (data a definir) | Dez-2016 (data a definir) |

Tabela 1: Calendarização do Projeto

7. Conclusões

7.1. *Objetivos atingidos*

Olhando para trás, especificamente para os objetivos que foram enunciados no início do projeto e que constam também do início deste relatório, boa parte dos mesmos aparenta ter sido atingida. Assim, foram atingidos os seguintes objetivos:

- “*Envio de e-mail por SMTP com cifra do canal de transporte.*”
- “*Receção de e-mail via IMAPS (Internet Message Access Protocol over SSL), igualmente com cifra do canal de transporte.*”
- “*Suporte a múltiplos domínios de e-mail*” - Neste caso, há que fazer uma ressalva: embora o sistema tenha ficado preparado para suportar múltiplos domínios de e-mail, apenas foi registado um domínio (uabdemo.info) que foi o único usado para os testes. Esta restrição está ligada a limitações de tempo (que estariam relacionados com as configurações e testes correspondentes) e também para evitar mais encargos financeiros decorrentes do registo de domínios adicionais.
- “*Suporte a múltiplas mailboxes e aliases (permitindo reencaminhamento de mensagens para domínios internos ou externos), com informação das mailboxes e aliases registada em Base de Dados.*” - Também foi realizada a configuração necessária, embora não testando a funcionalidade de *aliases/reencaminhamento*.
- “*Interface web (Webmail) para leitura e envio de e-mail, com certificado digital ativo para que as comunicações sejam feitas por HTTPS (Hyper Text Transfer Protocol Secure) para identificação do servidor e para cifrar o canal de transporte.*” - Neste caso, foi usada a *Certificate Authority da StartSSL* (<https://www.startssl.com/>) para assinar o CSR (*Certificate Signing Request*) que foi gerado. Especificamente, foi escolhida a variante gratuita chamada “*StartSSL Free*”.

- “*Funcionalidade proposta pelo Professor Orientador: Criar uma funcionalidade de reporting automático no sistema de email que permita monitorizar entregas de documentos como resposta a um pedido. A ideia seria, partindo de um e-mail enviado a um grupo de pessoas, que o sistema permita criar um relatório em qualquer momento, que indique o número e percentagem de respostas recebidas, com e sem ficheiros anexos. Esse relatório deveria também indicar quem do grupo de destinatários ainda não respondeu ao pedido (e eventualmente permitir a criação de um email de reforço apenas para esses).*” - Este objetivo foi conseguido, embora se deva reconhecer que com várias limitações: na configuração atual, é necessário a execução manual do programa Perl para refrescar a Base de Dados; a *interface web* não tem quaisquer elementos de *design* gráfico, não é interativa e não está integrada com o Webmail; não foram testados casos especiais de *mailings* com vários indicadores/prefixos de resposta no início do assunto; não foi verdadeiramente tratada a possibilidade de haver *mailings* diferentes com o mesmo assunto, nem a possibilidade de um mesmo utilizador ter vários endereços de e-mail. No entanto, a informação pretendida é devidamente compilada e apresentada na página, permitindo uma gestão mais eficaz dos pedidos e respetivas respostas efetuadas por *email*.

7.2. Oportunidades futuras de melhoria

Existem várias oportunidades de melhoria (algumas delas relacionadas com objectivos definidos no início do projeto, enquanto outras corresponderiam a novos objectivos). Assim, como trabalho futuro, poderia pensar-se em:

- Automatizar a instalação e/ou configuração de alguns dos componentes de *software* do sistema.
- Registar vários domínios e testar o sistema com vários domínios.
- Definir e testar *aliases / forwarding*.

- Configurar um sistema de anti-spam como o SpamAssassin, incorporando também filtros *bayesianos*.
- Instalar e configurar um *software* anti-vírus no servidor de e-mail (como, por exemplo, o ClamAV) integrando-o com o Postfix
- Implementar o backup via "mysqldump" para garantir que os backups são consistentes, no que diz respeito às Bases de Dados (quer da Base de Dados "dbvmail", quer da Base de Dados "dbmailings").
- Relativamente à funcionalidade proposta pelo Professor Orientador, existem várias oportunidades para trabalhos futuros:
 - ◊ Integrar, de alguma forma, o servidor de e-mail com o programa Perl criado de forma a que a receção de uma mensagem na caixa de mailings invoque automaticamente o programa Perl para refrescar a informação na Base de Dados.
 - ◊ Mover, para fora da "INBOX" e para uma outra pasta da *mailbox de mailings*, as mensagens que já tivessem sido guardadas na Base de Dados MySQL de *mailings*, de forma a reduzir o volume de mensagens que o programa tem de processar em cada execução. Nesse cenário alternativo, só teriam de ser processadas, em cada execução, as mensagens cuja meta-informação não tivesse ainda sido guardada na Base de Dados.
 - ◊ Criar uma *interface web* com *design* gráfico apelativo e funcionalidades mais interativas e, se possível, integrá-lo com o próprio sistema de Webmail.
 - ◊ Testar e tratar situações de mensagens com assuntos que tenham mais do que um indicador/prefixo de resposta no início do assunto.
 - ◊ Suportar a possibilidade de uma mesma pessoa ter vários endereços de e-mail.

8. Software utilizado neste Relatório

Para a elaboração deste relatório foi utilizado o seguinte ambiente:

- Sistema Operativo: *Distribuição Linux – Ubuntu MATE 16.04.1 LTS (Xenial Xerus) de 64-bits*
- Processador de Texto: *LibreOffice Writer 5.1.4.2*
- Software de Folha de Cálculo: *LibreOffice Calc 5.1.4.2*
- Editor de diagramas: *Dia 0.97.3*
- Editor de imagens: *Pinta 1.6*
- Serviço online de syntax highlighting: <https://tohtml.com/>

9. Referências Bibliográficas

- Bradner, S. (1997). RFC 2119 – *Key words for use in RFCs to Indicate Requirement Levels*. IETF, março 1997. <https://tools.ietf.org/html/rfc2119>
- Coulouris, G. et al. (2011). *Distributed Systems: Concepts and Design*. 5th Edition, Addison Wesley Longman.
- Crocker, D. (1982). RFC 822 – *Standard for the format of ARPA Internet Text Messages*. IETF, agosto 1982. <https://tools.ietf.org/search/rfc822>
- Kitterman, S. (2014). RFC 7208 – *Sender Policy Framework (SPF) for Authorizing Use of Domains in Email, Version 1*. IETF, abril 2014. <https://tools.ietf.org/html/rfc7208>
- Mockapetris, P. (1987a). RFC 1034 – *Domain Names – Concepts and Facilities*. IETF, novembro 1987. <https://tools.ietf.org/html/rfc1034>
- Mockapetris, P. (1987b). RFC 1035 – *Domain Names – Implementation and Specification*. IETF, novembro 1987. <https://tools.ietf.org/html/rfc1035>
- Resnick, P. (2001). RFC 2822 – Internet Message Format. IETF, abril 2001. <https://tools.ietf.org/search/rfc2822>
- Resnick, P. (2008). RFC 5322 – Internet Message Format. IETF, outubro 2008. <https://tools.ietf.org/search/rfc5322>
- Svistunov, M. et al. (2016). *Red Hat Enterprise Linux 6 Deployment Guide - Deployment, Configuration and Administration of Red Hat Enterprise Linux 6*. Red Hat, 2016. https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/6/html/Deployment_Guide/
- Wong, M. & Schlitt, W. (2006). RFC 4408 – *Sender Policy Framework (SPF) for Authorizing Use of Domains in Email, Version 1*. IETF, abril 2006. <https://tools.ietf.org/html/rfc4408>

10. Anexos

10.1. Ficheiro Perl

“messages_metadata_parser_to_db.pl”

```
#!/usr/bin/perl

use strict;
use warnings;

# Script "messages_metadata_parser_to_db.pl"
#
# Processes messages saved on an IMAP Server and saves meta-information
# to a MySQL Database
#
# Created by Ricardo Marques <1100281@estudante.uab.pt> on November 2016

# References consulted regarding IMAPClient:
# http://search.cpan.org/~plobbes/Mail-IMAPClient-
# 3.38/lib/Mail/IMAPClient.pod
# and
# http://stackoverflow.com/questions/11278922/trying-to-parse-text-and-
# html-from-email-over-imap-using-mailimapclient-but-te

use Mail::IMAPClient;
use MIME::EncWords qw(:all);      # for decoding MIME e-mail subjects
use Email::Address;
use Email::MIME::Attachment::Stripper; # for handling e-mail attachments
use DBI;    # for connecting to Databases and working with them
use constant MAILINGS_EMAIL_ADDRESS => qw(mailings@uabdemos.info);

my $imap = Mail::IMAPClient->new(
    Server  => 'localhost',
    User    => MAILINGS_EMAIL_ADDRESS,
    Password => '[Password do utilizador]',
    Ssl     => 0,
    Uid     => 0,
);

# Function is_mailing
# Checks if a message is the original message of a Mailing
# Returns true (1) if a message is the original message of a Mailing
```

```
# Returns false (0) if a message is a Reply to a Mailing
sub is_mailing {
    my ($subject_decoded) = @_;
    my $is_mailing = 1;

    # If the subject does NOT start with "Re: " OR "RE: ",
    # then it is NOT a mailing:
    if ($subject_decoded =~ /^R[Ee]: ?/) {
        $is_mailing = 0;
    }

    return($is_mailing);
} # END is_mailing

# Function to remove the prefix of a Reply ("Re: " or "RE: ")
# from the beginning of the subject
sub trim_subject_from_re {
    my ($subject_decoded) = @_;
    my $subject_trimmed_from_re = $subject_decoded;
    $subject_trimmed_from_re =~ s/^R[Ee]: ?//;

    return($subject_trimmed_from_re);
} # END trim_subject_from_re

# Function is_message_in_database
# Checks if a message is already in the database
# Returns true (1) if a message is already in the Messages table
# Returns false (0) if a message is not in the Messages table
sub is_message_in_database {
    my ($message_id, $dbh) = @_;
    my $is_message_in_database = 0;

    my $query = "SELECT message_id FROM Messages WHERE message_id LIKE '%
$message_id%';";
    # Check if message-id exists in Database.
    my $th = $dbh->prepare($query);
    $th->execute();

    if ($th->rows > 0) {
        $is_message_in_database = 1;
    }
}
```

```
    return ($is_message_in_database);

} # END is_message_in_database

# Function is_recipient_the_mailing_sender
# Returns true (1) if the recipient is the mailing sender for the given
# subject -AND- fills the mailing message id as a side effect
# Returns false (0) if the recipient is NOT the mailing sender
sub is_recipient_the_mailing_sender {
    my ($recipient_email_address, $subject_trimmed_from_re,
$mailing_message_id_ref, $dbh) = @_;

    my $is_recipient_the_mailing_sender = 0;

    my $query = "SELECT message_id FROM Messages WHERE
sender_email_address LIKE '$recipient_email_address' "
        "AND subject LIKE '$subject_trimmed_from_re' AND
is_mailing=1;";

    # Check if message-id exists in Database where the recipient is the
    # mailing sender
    my $th = $dbh->prepare("$query");
    $th->execute();

    if ($th->rows > 0) {
        $is_recipient_the_mailing_sender = 1;
        $mailing_message_id_ref = $th->fetchrow_array();
    }

    return ($is_recipient_the_mailing_sender);
} # END is_recipient_the_mailing_sender

# Function is_mailing_fully_replied
sub is_mailing_fully_replied {
    my ($mailing_message_id, $dbh) = @_;

    my $is_mailing_fully_replied = 1;

    # Check if there is any recipient that has not replied the Mailing
    my $query = "SELECT message_id FROM Messages_Recipients WHERE
message_id LIKE '$mailing_message_id' "
        "AND has_been_replied_by_recipient != 1 ;";

    my $th = $dbh->prepare("$query");
    $th->execute();
```

```
if ($th->rows > 0) {
    $is_mailing_fully_replied = 0;
}

return ($is_mailing_fully_replied);
} # END is_mailing_fully_replied

# Function is_person_in_database
# Checks if a person is already in the database
# Returns true (1) if a person is already in the People table
# Returns false (0) if a person is not in the People table
sub is_person_in_database {
    my ($email_address, $dbh) = @_;

    my $is_person_in_database = 0;

    my $query = "SELECT email_address FROM People WHERE email_address
LIKE '%$email_address%';";

    # Check if message-id exists in Database. If it does not exist, let
    us add it:
    my $th = $dbh->prepare("$query");
    $th->execute();

    if ($th->rows > 0) {
        $is_person_in_database = 1;
    }

    return ($is_person_in_database);
} # END is_person_in_database

# Procedure insert_message_in_database
# Inserts a message in the Messages table of the Mailings database
sub insert_message_in_database {
    my ($message_id, $from_email_address, $subject, $dbh) = @_;

    my $is_mailing = 0;

    if (is_mailing($subject)) {
        $is_mailing = 1;
    } else {
        # This is not a Mailing, but a reply to a Mailing.
        # Let's remove the "Re:" from the beginning of the subject
        $subject = trim_subject_from_re($subject);
    }
}
```

```
}

my $query = "INSERT INTO Messages (message_id, sender_email_address,
is_mailing, subject) "
          "VALUES ('$message_id', '$from_email_address',
$is_mailing, '$subject');";
#print "QUERY = $query\n";
my $th = $dbh->prepare(qq{$query});
$th->execute();

} # END insert_message_in_database

# Procedure insert_person_in_database
# Inserts a person in the People table of the Mailings database
sub insert_person_in_database {
    my ($to_email_address, $to_email_name, $dbh) = @_;

    # There is not a record for this person in the People table. Let's
    # insert the record:
    my $query = "INSERT INTO People (email_address, name) "
               "VALUES ('$to_email_address', '$to_email_name');";

    my $th = $dbh->prepare(qq{$query});
    $th->execute();

} # END insert_person_in_database

# Procedure insert_message_recipient_pair_in_database
# Inserts a message-recipient pair in the Messages_Recipients table of
# the Mailings database
sub insert_message_recipient_pair_in_database {
    my ($message_id, $recipient_email_address, $dbh) = @_;

    # Let's insert the related record for the Messages_Recipients table:
    my $query = "INSERT INTO Messages_Recipients (message_id,
recipient_email_address) "
               "VALUES ('$message_id', '$recipient_email_address');";

    my $th = $dbh->prepare(qq{$query});
    $th->execute();

} # END insert_message_recipient_pair_in_database

# Procedure insert_attachment_information_in_database
```

```
# - Updates the "has attachments" flag in the Messages table
# - Inserts the filenames of the attachments in the Messages_Attachments
#   table
sub insert_attachment_information_in_database {
  my ($message_id, $message_ref, $dbh_ref) = @_;
  my $message = $$message_ref;
  my $dbh = $$dbh_ref;
  my $has_attachments = 0;
  my $query = "";
  # NAIVE approach: get lines that have filename="file name"
  # (I am doing this because Email::MIME::Attachment::Stripper was
  failing me)
  # The following loads the WHOLE message (including attachments) to a
  variable
  # This obviously is memory-intensive (the larger the attachments are,
  # the more memory gets used)
  my $message_string = $imap->message_string($message);
  # Let's treat the string as if it was a file
  # ( Idea taken from:
  #     Parsing multiline string line by line
  #     http://www.perlmonks.org/?node\_id=745018 ):
  open my $fh, '<', \$message_string;
  while (my $line = <$fh>) {
    if ($line =~ /filename="(.*)"/) {
      if (!$has_attachments) {
        $has_attachments = 1;
        $query = "UPDATE Messages SET has_attachments=1 WHERE
message_id LIKE '%$message_id%'";
        #print "QUERY = $query\n";
        my $sth = $dbh->prepare(qq{$query});
        $sth->execute();
      }
      my $filename = $1;
      #print "Matched: $filename\n";
      $query = "INSERT INTO Messages_Attachments (message_id,
filename) VALUES('$message_id', '$filename');";
      my $sth = $dbh->prepare(qq{$query});
      $sth->execute();
    }
  }
  close $fh;
} # END insert_attachment_information_in_database

# Procedure set_replied_flag_to_true_in_database
sub set_replied_flag_to_true_in_database {
```

```
my($mailing_message_id, $recipient_email_address, $dbh) = @_;

my $query ="UPDATE Messages_Recipients SET
has Been_Replied_By_Recipient=1"
    " WHERE message_id LIKE '$mailing_message_id' AND
recipient_email_address LIKE '$recipient_email_address';";
#print "Query = $query\n";

my $th = $dbh->prepare(qq{$query});
$th->execute();

} # END set_replied_flag_to_true_in_database

# Procedure set_fully_replied_flag_to_true_in_database
sub set_fully_replied_flag_to_true_in_database {
    my($mailing_message_id, $dbh) = @_;

    my $query ="UPDATE Messages SET has_Been_Fully_Replied_If_Mailing=1 "
    " WHERE message_id LIKE '$mailing_message_id' ;";
#print "Query = $query\n";

    my $th = $dbh->prepare(qq{$query});
    $th->execute();

} # END set_fully_replied_flag_to_true_in_database

# Procedure parse_messages
sub parse_messages {
    my ($messages_ref, $dbh_ref) = @_;
    my @messages = @{$messages_ref}; # Using @ to dereference a
reference-to-an-array
    my $dbh = $$dbh_ref; # The $$ operator serves to dereference a
reference
    my $message_counter = 0;

    my $query = "";

    foreach my $message (@messages) {

        print "=" x 40;
        print "\n";

        $message_counter++;
        print "Message #$message_counter\n\n";
    }
}
```

```
my $message_id_with_angle_brackets = $imap->get_header($message, "Message-Id" );
# Remove the first and last characters to get rid of the angle brackets:
my $message_id = substr($message_id_with_angle_brackets, 1, -1);

my $from = $imap->get_header( $message, "From" );
my $from_decoded = decode_mimewords($from);
my @from_email_addresses = Email::Address->parse($from_decoded);
my $from_email_address = $from_email_addresses[0]->address;
my $return_path = $imap->get_header( $message, "Return-Path" );

my $to = $imap->get_header( $message, "To" );
my $to_decoded = decode_mimewords($to);

my $cc = $imap->get_header ($message, "Cc" );
my $cc_decoded = decode_mimewords($cc);

my $subject = $imap->get_header( $message, "Subject" );
my $subject_decoded = decode_mimewords($subject);
my $date = $imap->get_header( $message, "Date" );

if (!is_message_in_database($message_id, $dbh)) {
    # There is NOT yet a record for this message ID. Let's insert the record:
    insert_message_in_database($message_id, $from_email_address, $subject_decoded, $dbh);
    my $subject_trimmed_from_re =
trim_subject_from_re($subject_decoded);

    my @to_email_addresses_with_names = Email::Address->parse($to_decoded);
    my @cc_email_addresses_with_names = Email::Address->parse($cc_decoded);

    # Let's fill an array of recipients with the "To" and "Cc" recipients arrays
    my @recipients_email_addresses_with_names;
    push(@recipients_email_addresses_with_names,
@to_email_addresses_with_names);
    push(@recipients_email_addresses_with_names,
@cc_email_addresses_with_names);

    # Handle Recipients:
    foreach my $recipient_email_address_with_names
(@recipients_email_addresses_with_names) {
        my $recipient_email_name =
$recipient_email_address_with_names->name;
        my $recipient_email_address =
$recipient_email_address_with_names->address;
```

```
#print "Recipient email name: $recipient_email_name\n";
#print "Recipient email address:
$recipient_email_address\n";

# One of the recipients will be the mailings e-mail
address
# We'll only consider the cases of other recipients
if ($recipient_email_address ne MAILINGS_EMAIL_ADDRESS) {
    if (!is_person_in_database($recipient_email_address,
$dbh)) {
        # There is NOT yet a record for this person.
Let's insert the record:

insert_person_in_database($recipient_email_address,
$recipient_email_name, $dbh);
}

insert_message_recipient_pair_in_database($message_id,
$recipient_email_address, $dbh);

# Check if the recipient is the sender of the
original mailing
# If it is, let's update the "has been replied" flag
to TRUE
my $mailing_message_id;
# We'll pass $mailing_message_id to fill that
variable by reference as a side effect
if
(is_recipient_the_mailing_sender($recipient_email_address,
$subject_trimmed_from_re, \$mailing_message_id, $dbh)) {

set_replied_flag_to_true_in_database($mailing_message_id,
$from_email_address, $dbh);
    if(is_mailing_fully_replied($mailing_message_id,
$dbh)) {

set_fully_replied_flag_to_true_in_database($mailing_message_id, $dbh);
    }
}
}

# Handle Attachments:
insert_attachment_information_in_database($message_id, \
$message, \$dbh);

} #else {
#    print "There were no records for Message-ID =
$message_id\n";
```

```
#}

} # END of parse_messages

# MAIN

# Connect to MySQL Database
my $db_username = 'mailings_dbuser';
my $db_password = '[Password do Utilizador]';
my $db_server = 'localhost';
my $db_name = 'dbmailings';

my $dbh = DBI->connect("DBI:mysql:$db_name:$db_server", $db_username,
$db_password)
    or die("ERROR connecting to the Database");

my $folders;

if (defined ($imap->folders)) {
    $folders = $imap->folders
        or die "List folders error: " . $imap->LastError, "\n";
}

#print "Folders: @$folders\n";

my %Opt;
$Opt{folder} = "INBOX";

$imap->select( $Opt{folder} )
    or die("Select " . $Opt{folder} . "error: " . $imap->LastError .
"\n");

my @messages = $imap->messages;

parse_messages(\@messages, \$dbh); # The "\\" (backslash) serves to pass
references

# Close connection to IMAP Server:
$imap->logout
    or die "Logout error: " . $imap->LastError, "\n";

# Close connection to database:
$dbh->disconnect();
```

10.2. Ficheiro PHP “mailings.php”

```
<?php

/* Script "mailings.php"
 *
 * Displays on a web page the meta-information regarding a Mailings
database
 * that is stored on a MySQL database.
 *
 * Created by Ricardo Marques <1100281@estudante.uab.pt> on November
2016
 */

// Reference consulted:
// http://php.net/manual/en/mysqli.quickstart.dual-interface.php

$db_username = 'mailings_dbuser';
$db_password = '[Password do Utilizador]';
$db_server = 'localhost';
$db_name = 'dbmailings';

$mysqli = new mysqli("$db_server", "$db_username", "$db_password",
"$db_name");
if ($mysqli->connect_errno) {
    echo "Failed to connect to MySQL: " . $mysqli->connect_error;
}

// Function write_people_table
function write_people_table() {
    Global $mysqli;
    $res = $mysqli->query("SELECT name, email_address FROM People;");
    echo "<table border=1>\n";
    echo "<tr><th>Name</th><th>Email address</th></tr>\n";

    while ($row = $res->fetch_array()) {
        echo "<tr><td>" . $row["name"] . "</td><td>" .
$row["email_address"] . "</td></tr>\n";
    }

    echo "</table>\n";
} // END write_people_table

// Function write_field
function write_field($label, $value) {
    echo "<strong>$label: </strong>" . $value . "<br />\n";
} // END write_field
```

```
// Function write_individual_attachments_info($message_id);
function write_individual_attachments_info($message_id) {
    Global $mysqli;

    $query = "SELECT message_id, filename FROM Messages_Attachments "
            " WHERE Messages_Attachments.message_id =
    '$message_id' ";

    $res = $mysqli->query("$query");

    $counter = 0;
    while ($row = $res->fetch_array()) {
        $counter++;
        write_field("Attachment #$counter in Reply",
    $row["filename"]);
    }

} // END write_individual_attachments_info

// Function write_attachments_info
function write_attachments_info($sender_of_reply_email_address,
$mailing_initiator_email_address, $subject) {
    Global $mysqli;

    $query = "SELECT Messages.message_id, Messages.has_attachments
FROM Messages_Recipients INNER JOIN Messages "
            " ON Messages_Recipients.message_id =
    Messages.message_id "
            " WHERE Messages.sender_email_address =
    '$sender_of_reply_email_address' "
            " AND Messages_Recipients.recipient_email_address =
    '$mailing_initiator_email_address' "
            " AND Messages.subject = '$subject';";

    #write_field("Query", $query);
    $res = $mysqli->query("$query");

    while ($row = $res->fetch_array()) {
        $has_attachments = $row["has_attachments"];
        if (!$has_attachments) {
            write_field("Has attachments", "No");
        } else {
            $message_id = $row["message_id"];
            write_field("Reply has attachments", "Yes");
        }
    }
}
```

```
        write_individual_attachments_info($message_id);
    }
}

} // END write_attachments_info

// Function write_recipients_info
function write_recipients_info($mailing_message_id) {
    Global $mysqli;

    $query = "SELECT People.name, sender_email_address,
recipient_email_address, subject, has_been_replied_by_recipient "
        " FROM Messages_Recipients "
        " INNER JOIN People INNER JOIN Messages "
        " ON Messages_Recipients.recipient_email_address =
People.email_address "
        " AND Messages.message_id =
Messages_Recipients.message_id "
        " WHERE Messages.message_id = '$mailing_message_id' ";
//write_field("Query:", $query);
$res = $mysqli->query("$query");

while ($row = $res->fetch_array()) {
    $has_been_replied_by_recipient =
$row["has_been_replied_by_recipient"];
    write_field("Name", $row["name"]);
    write_field("Has replied", ($has_been_replied_by_recipient ==
1 ?
                    "<span style=\"color:green\">Yes</span>" :
                    "<span style=\"color:red\">No</span>"));
    if ($has_been_replied_by_recipient) {
        // If recipient replied, let's write the information
about attachments on the reply
        $sender_of_reply_email_address =
$row["recipient_email_address"];
        $mailing_initiator_email_address =
$row["sender_email_address"];
        $subject = $row["subject"];
        //echo "Sender of reply:
$sender_of_reply_email_address<br />";
        //echo "Mailing initiator :
$mailing_initiator_email_address<br />";
        //echo "Subject: $subject<br />";
        write_attachments_info($sender_of_reply_email_address,
$mailing_initiator_email_address, $subject);
    }
    echo "<br />";
}
}
```

```
    } // END write_recipients_info

    // Function write_mailings_info
    function write_mailings_info() {
        Global $mysqli;
        $query = "SELECT name, subject, message_id,
has_been_fully_replied_if_mailing from Messages INNER JOIN People "
            " ON Messages.sender_email_address=People.email_address
        "
            " WHERE is_mailing=1 ";
        $res = $mysqli->query("$query");

        $counter = 0;
        while ($row = $res->fetch_array()) {
            $counter++;
            echo "<h3>- Mailing #". $counter . "</h3>";
            write_field("Mailing initiator", $row["name"]);
            write_field("Subject", $row["subject"]);
            write_field("Message ID", $row["message_id"]);
            write_field("Replied by all recipients",
            ($row["has_been_fully_replied_if_mailing"] == 1 ?
                "<span style=\"color:green\>Yes</span>" :
                "<span style=\"color:red\>No</span>"));

            echo "<h4>Recipients:</h4>\n";
            $mailing_message_id = $row["message_id"];

            write_recipients_info($mailing_message_id);

            echo "<hr />\n";
        }
    } // END write_mailings_info

?>

<html>
    <head>
        <title>Mailings Report</title>
    </head>

    <body>
        <h1>Mailings Report</h1>
        <h2>People</h2>
```

```
<?php write_people_table(); ?>

<h2>Mailings</h2>
<?php write_mailings_info(); ?>

</body>
</html>

<?php
    // Close connection to the Database:
    mysqli_close($mysql);
?>
```

10.3. Comandos executados na máquina virtual

10.3.1. Instalação e configuração do Apache

```
[root@machine2 ~]# yum install httpd

[root@machine2 ~]# yum install mod_ssl

[root@machine2 ~]# cd /etc/httpd/conf.d/

[root@machine2 conf.d]# vim ssl.conf

[root@machine2 conf.d]# cat ssl.conf
# RICARDO
# 8-Oct-2016

# Section created for SSL VirtualHosts (vhosts)

Listen 443
LoadModule ssl_module modules/mod_ssl.so

<VirtualHost 138.68.131.28:443>
    SSLEngine On
    SSLCertificateFile /etc/pki/tls/certs/www.uabdemo.info.crt
    SSLCertificateKeyFile /etc/pki/tls/private/www.uabdemo.info.key
    ServerName www.uabdemo.info
    DocumentRoot /var/www/vhosts/www.uabdemo.info/html
    <Directory /var/www/vhosts/www.uabdemo.info/html>
        AllowOverride All
    </Directory>
</VirtualHost>

[root@machine2 httpd]# chkconfig httpd on
[root@machine2 httpd]# service httpd restart
```

10.3.2. Instalação e configuração do MySQL

```
[root@machine2 ~]# yum install mysql-server
[root@machine2 ~]# service mysqld stop
[root@machine2 ~]# service mysqld start
[root@machine2 ~]# chkconfig mysqld on

[root@machine2 ~]# mysql_secure_installation
NOTE: RUNNING ALL PARTS OF THIS SCRIPT IS RECOMMENDED FOR ALL MySQL
      SERVERS IN PRODUCTION USE! PLEASE READ EACH STEP CAREFULLY!
In order to log into MySQL to secure it, we'll need the current
password for the root user. If you've just installed MySQL, and
you haven't set the root password yet, the password will be blank,
so you should just press enter here.

Enter current password for root (enter for none):
OK, successfully used password, moving on...

Setting the root password ensures that nobody can log into the MySQL
root user without the proper authorisation.

Set root password? [Y/n] Y
New password:
Re-enter new password:
Password updated successfully!
Reloading privilege tables..
... Success!

By default, a MySQL installation has an anonymous user, allowing anyone
to log into MySQL without having to have a user account created for
them. This is intended only for testing, and to make the installation
go a bit smoother. You should remove them before moving into a
production environment.

Remove anonymous users? [Y/n] Y
... Success!

Normally, root should only be allowed to connect from 'localhost'. This
ensures that someone cannot guess at the root password from the network.

Disallow root login remotely? [Y/n] Y
... Success!

By default, MySQL comes with a database named 'test' that anyone can
access. This is also intended only for testing, and should be removed
before moving into a production environment.

Remove test database and access to it? [Y/n] Y
```

```
- Dropping test database...
... Success!
- Removing privileges on test database...
... Success!
```

Reloading the privilege tables will ensure that all changes made so far will take effect immediately.

```
Reload privilege tables now? [Y/n] Y
... Success!
```

Cleaning up...

All done! If you've completed all of the above steps, your MySQL installation should now be secure.

Thanks for using MySQL!

Para a criação da Base de Dados “dbvmail” associada aos domínios de e-mail, *mailboxes* e *aliases*, foram usados, como base / referência, os seguintes tutoriais:

<https://www.digitalocean.com/community/tutorials/how-to-configure-a-mail-server-using-postfix-dovecot-mysql-and-spamassassin>

<https://www.linode.com/docs/email/postfix/email-with-postfix-dovecot-and-mysql>

Os comandos conducentes à criação do utilizador associado a essa Base de Dados e à criação da própria Base de Dados foram os seguintes:

```
[root@machine2 /]# mysqladmin -p create dbvmail
[root@machine2 /]# mysql -u root -p
mysql> GRANT SELECT ON dbvmail.* TO 'vmail'@'127.0.0.1' IDENTIFIED BY
'[Password do Utilizador]';
Query OK, 0 rows affected (0.00 sec)

mysql> GRANT SELECT ON dbvmail.* TO 'vmail'@'localhost' IDENTIFIED BY
'[Password do Utilizador]';
Query OK, 0 rows affected (0.00 sec)

mysql> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.00 sec)

mysql> USE dbvmail;
Database changed
```

```
mysql> CREATE TABLE `virtual_domains` (
-> `id` INT NOT NULL AUTO_INCREMENT,
-> `name` VARCHAR(50) NOT NULL,
-> PRIMARY KEY (`id`)
-> ) ENGINE=InnoDB DEFAULT CHARSET=utf8;
Query OK, 0 rows affected (0.04 sec)

mysql> CREATE TABLE `virtual_aliases` (
-> `id` INT NOT NULL AUTO_INCREMENT,
-> `domain_id` INT NOT NULL,
-> `source` varchar(100) NOT NULL,
-> `destination` varchar(100) NOT NULL,
-> PRIMARY KEY (`id`),
-> FOREIGN KEY (domain_id) REFERENCES virtual_domains(id) ON DELETE
CASCADE
-> ) ENGINE=InnoDB DEFAULT CHARSET=utf8;
Query OK, 0 rows affected (0.00 sec)

mysql> CREATE TABLE `virtual_users` (
-> `id` INT NOT NULL AUTO_INCREMENT,
-> `domain_id` INT NOT NULL,
-> `password` VARCHAR(106) NOT NULL,
-> `email` VARCHAR(120) NOT NULL,
-> PRIMARY KEY (`id`),
-> UNIQUE KEY `email` (`email`),
-> FOREIGN KEY (domain_id) REFERENCES virtual_domains(id) ON DELETE
CASCADE
-> ) ENGINE=InnoDB DEFAULT CHARSET=utf8;
Query OK, 0 rows affected (0.01 sec)

mysql> INSERT INTO `dbvmail`.`virtual_domains` (`id` , `name` ) VALUES
('1', 'uabdemo.info');
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO `dbvmail`.`virtual_users`(`id`, `domain_id`, `password` , `email` )
VALUES ('1', '1', ENCRYPT('[Password do Utilizador]', CONCAT('$6$', SUBSTRING(SHA(RAND()), -16))), 'ricardo.marques@uabdemo.info');
```

10.3.3. Comandos para gerar o par de chave privada e pública RSA do certificado

```
[root@machine2 ~]# openssl genrsa -out uabdemoinfo.key 2048
[root@machine2 ~]# chmod 600 uabdemoinfo.key
```

10.3.4. Comandos para gerar o CSR (*Certificate Signing Request*) do Certificado

```
[root@machine2 ~]# openssl req -new -key uabdemoinfo.key -out uabdemoinfo.csr
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a
DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [XX]:PT
State or Province Name (full name) []:Lisboa
Locality Name (eg, city) [Default City]:Lisboa
Organization Name (eg, company) [Default Company Ltd]:UAbDemo
Organizational Unit Name (eg, section) []:
Common Name (eg, your name or your server's hostname) []:uabdemo.info
Email Address []:domains@ricmarques.net

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:
```

O CSR (*Certificate Signing Request*) correspondente foi submetido na *interface Web* do site da *StartSSL* e os restantes *SAN* (*Subject Alternative Names*) do certificado foram adicionados, através dessa *interface Web*, no processo de assinatura do certificado pela CA (*Certificate Authority*) da *StartSSL*. Os nomes (SAN) adicionados ao certificado foram os seguintes:

- www.uabdemo.info
- webmail.uabdemo.info
- mail.uabdemo.info
- machine1.uabdemo.info
- machine2.uabdemo.info

10.3.5. Instalação e configuração do Postfix

A versão do Postfix que vem com o repositório do CentOS não inclui suporte para MySQL. Por isso, é desinstalada a versão que vem instalada e é instalada a versão que está disponível no repositório CentOSPlus que, essa sim, inclui suporte para MySQL.

Para a instalação e configuração do Postfix, foram usados, como base / referência, os seguintes tutoriais:

<https://www.digitalocean.com/community/tutorials/how-to-configure-a-mail-server-using-postfix-dovecot-mysql-and-spamassassin>

<https://www.linode.com/docs/email/postfix/email-with-postfix-dovecot-and-mysql>

```
[root@machine2 yum.repos.d]# yum remove postfix
[root@machine2 yum.repos.d]# cp -p CentOS-Base.repo CentOS-Base.repo.ORIG.20161009

[root@machine2 yum.repos.d]# diff CentOS-Base.repo.ORIG.20161009 CentOS-Base.repo
18a19,26
> # RICARDO
> # 9-oct-2016
> # Based on the following article:
> # https://www.linode.com/docs/email/postfix/email-with-postfix-dovecot-and-mysql-on-centos-6
> # ... I am excluding "postfix" from the Base and Updates repositories,
because we will use
> # the "postfix" package from the "CentOS Plus" repository because that
other package has
> # MySQL support that we need
> exclude=postfix
(...)

[root@machine2 yum.repos.d]# yum --enablerepo=centosplus install postfix
[root@machine2 ~]# cd /etc/postfix/
[root@machine2 postfix]# vim mysql-virtual-mailbox-domains.cf
[root@machine2 postfix]# cat mysql-virtual-mailbox-domains.cf
user = vmail
password = [Password do utilizador]
```

```
hosts = 127.0.0.1
dbname = dbvmail
query = SELECT 1 FROM virtual_domains WHERE name='%s'

[root@machine2 postfix]# vim mysql-virtual-mailbox-maps.cf

[root@machine2 postfix]# cat mysql-virtual-mailbox-maps.cf
user = vmail
password = [Password do utilizador]
hosts = 127.0.0.1
dbname = dbvmail
query = SELECT 1 FROM virtual_users WHERE email='%s'

[root@machine2 postfix]# vim mysql-virtual-alias-maps.cf

[root@machine2 postfix]# cat mysql-virtual-alias-maps.cf
user = vmail
password = [Password do utilizador]
hosts = 127.0.0.1
dbname = dbvmail
query = SELECT destination FROM virtual_aliases WHERE source='%s'

[root@machine2 postfix]# cp -p master.cf master.cf.ORIG.20161013

[root@machine2 postfix]# vim master.cf

[root@machine2 postfix]# diff master.cf.ORIG.20161013 master.cf
12,21c12,28
< #submission inet n      -      n      -      -      smtpd
< #  -o smtpd_tls_security_level=encrypt
< #  -o smtpd_sasl_auth_enable=yes
< #  -o smtpd_client_restrictions=permit_sasl_authenticated,reject
< #  -o milter_macro_daemon_name=ORIGINATING
< #smtps      inet n      -      n      -      -      smtpd
< #  -o smtpd_tls_wrappermode=yes
< #  -o smtpd_sasl_auth_enable=yes
< #  -o smtpd_client_restrictions=permit_sasl_authenticated,reject
< #  -o milter_macro_daemon_name=ORIGINATING
---
>
> # RICARDO
> # 13-oct-2016
> # Uncommented the following block according to
> # https://www.linode.com/docs/email/postfix/email-with-postfix-
dovecot-and-mysql
> submission inet n      -      n      -      -      smtpd
>   -o smtpd_tls_security_level=encrypt
>   -o smtpd_sasl_auth_enable=yes
>   -o smtpd_client_restrictions=permit_sasl_authenticated,reject
>   -o milter_macro_daemon_name=ORIGINATING
> smtps      inet n      -      n      -      -      smtpd
>   -o smtpd_tls_wrappermode=yes
```

```
> -o smtpd_sasl_auth_enable=yes
> -o smtpd_client_restrictions=permit_sasl_authenticated,reject
> -o milter_macro_daemon_name=ORIGINATING
> # END of uncommented block
> #

[root@machine2 ~]# cd /etc/
[root@machine2 etc]# chmod -R o-rwx /etc/postfix/
[root@machine2 etc]# service postfix restart
```

10.3.6. Instalação e configuração do Dovecot

Para a instalação e configuração do Dovecot, foram usados, como base / referência, os seguintes tutoriais:

<https://www.digitalocean.com/community/tutorials/how-to-configure-a-mail-server-using-postfix-dovecot-mysql-and-spamassassin>

<https://www.linode.com/docs/email/postfix/email-with-postfix-dovecot-and-mysql>

Os comandos usados para a instalação e configuração do Dovecot foram os seguintes:

```
[root@machine2 ~]# yum install dovecot
[root@machine2 ~]# cd /etc/dovecot
[root@machine2 dovecot]# cp -p dovecot.conf dovecot.conf.ORIG.20161013
[root@machine2 dovecot]# diff dovecot.conf.ORIG.20161013 dovecot.conf
20c20,25
< #protocols = imap pop3 lmtp
---
> # RICARDO
> # 13-oct-2016
> # Uncommented the following line
> # based on:
> # https://www.linode.com/docs/email/postfix/email-with-postfix-dovecot-and-mysql
> protocols = imap pop3 lmtp
[root@machine2 dovecot]# cd conf.d/
```

```
[root@machine2 conf.d]# cp -p 10-mail.conf 10-mail.conf.ORIG.20161013
[root@machine2 conf.d]# diff 10-mail.conf.ORIG.20161013 10-mail.conf
31a32,38
> # RICARDO
> # 13-oct-2016
> # Added the following line based on
> # https://www.linode.com/docs/email/postfix/email-with-postfix-
dovecot-and-mysql
> mail_location = maildir:/var/mail/vhosts/%d/%n
>
>
115a123,128
> #
> # RICARDO
> # 13-oct-2016
> # Added the following line based on
> # https://www.linode.com/docs/email/postfix/email-with-postfix-
dovecot-and-mysql
> mail_privileged_group = mail

[root@machine2 conf.d]# cp -p 10-auth.conf 10-auth.conf.ORIG.20161013
[root@machine2 conf.d]# diff 10-auth.conf.ORIG.20161013 10-auth.conf
9c9,15
< #disable_plaintext_auth = yes
---
>
> # RICARDO
> # 13-oct-2016
> # Uncommented the following line
> # based on
> # https://www.linode.com/docs/email/postfix/email-with-postfix-
dovecot-and-mysql
> disable_plaintext_auth = yes
97c103,111
< auth_mechanisms = plain
---
> #
> # RICARDO
> # 13-oct-2016
> # Based on
> # https://www.linode.com/docs/email/postfix/email-with-postfix-
dovecot-and-mysql
> # ... I changed the following line from:
> #auth_mechanisms = plain
> # to:
> auth_mechanisms = plain login
119,120c133,141
< !include auth-system.conf.ext
< #!include auth-sql.conf.ext
```

```
---
> # RICARDO
> # 13-oct-2016
> # Based on
> # https://www.linode.com/docs/email/postfix/email-with-postfix-
dovecot-and-mysql
> # I commented the following line:
> #!/include auth-system.conf.ext
> # ...and uncommented the following line:
> !include auth-sql.conf.ext
>

[root@machine2 conf.d]# cp -p 10-master.conf 10-
master.conf.ORIG.20161013

[root@machine2 conf.d]# vim 10-master.conf

[root@machine2 conf.d]# diff 10-master.conf.ORIG.20161013 10-master.conf
19c19,23
<      #port = 143
---
>      port = 143
>      # Added the following line
>      # based on
>      # https://www.digitalocean.com/community/tutorials/how-to-
configure-a-mail-server-using-postfix-dovecot-mysql-and-spamassassin
>      #port=0
22c26,29
<      #port = 993
---
>      # RICARDO
>      # 16-oct-2016
>      # Uncommented the following lines:
>      port = 993
49c56,66
<      unix_listener lmtp {
---
>      # RICARDO
>      # 15-oct-2016
>      # Based on
>      # https://www.digitalocean.com/community/tutorials/how-to-configure-
a-mail-server-using-postfix-dovecot-mysql-and-spamassassin
>      # I commented out the following line
>      # unix_listener lmtp {
>      # ... and replaced it by:
>      unix_listener /var/spool/postfix/private/dovecot-lmtp {
>          mode = 0600
>          user = postfix
>          group = postfix
75a93,104
>      # RICARDO
```

```
>      # 15-oct-2016
>      # Based on
>      # https://www.digitalocean.com/community/tutorials/how-to-configure-
a-mail-server-using-postfix-dovecot-mysql-and-spamassassin
>      # I added the following block:
>      unix_listener /var/spool/postfix/private/auth {
>          mode = 0666
>          user = postfix
>          group = postfix
>      }
>
>
84a114,120
>      # RICARDO
>      # 15-oct-2016
>      # Based on
>      # https://www.digitalocean.com/community/tutorials/how-to-
configure-a-mail-server-using-postfix-dovecot-mysql-and-spamassassin
>      # I added the following lines:
>      mode = 0600
>      user = vmail
93a130,136
>
>      # RICARDO
>      # 15-oct-2016
>      # Based on
>      # https://www.digitalocean.com/community/tutorials/how-to-configure-
a-mail-server-using-postfix-dovecot-mysql-and-spamassassin
>      # I added the following lines:
>      user = dovecot
100a144,149
>      # RICARDO
>      # 15-oct-2016
>      # Based on
>      # https://www.digitalocean.com/community/tutorials/how-to-configure-
a-mail-server-using-postfix-dovecot-mysql-and-spamassassin
>      # I added the following lines:
>      user = vmail

[root@machine2 conf.d]# cp -p 10-ssl.conf 10-ssl.conf.ORIG.20161013

[root@machine2 conf.d]# diff /root/10-ssl.conf.ORIG.20161013 10-ssl.conf
6c6,12
< #ssl = yes
---
> ssl = yes

24a31,34
> # RICARDO
> # 16-Oct-2016
> # Added the following line
> ssl_ca = </etc/pki/tls/certs/ca-bundle.crt
```

```
[root@machine2 dovecot]# cd conf.d/
[root@machine2 conf.d]# mkdir -p /var/mail/vhosts/uabdemo.info
[root@machine2 conf.d]# groupadd -g 5000 vmail
[root@machine2 conf.d]# useradd -g vmail -u 5000 vmail -d /var/mail/
useradd: warning: the home directory already exists.
Not copying any file from skel directory into it.

[root@machine2 conf.d]# chown -R vmail:vmail /var/mail
[root@machine2 conf.d]# vim auth-sql.conf.ext

[root@machine2 conf.d]# diff auth-sql.conf.ext.ORIG.20161013 auth-
sql.conf.ext
19,22c19,27
< userdb {
<   driver = sql
<   args = /etc/dovecot/dovecot-sql.conf.ext
< }
---
> # RICARDO
> # 13-oct-2016
> # Based on
> # https://www.linode.com/docs/email/postfix/email-with-postfix-
dovecot-and-mysql
> # I commented the following block:
> #userdb {
> #   driver = sql
> #   args = /etc/dovecot/dovecot-sql.conf.ext
> #}
23a29,32
> # RICARDO
> # 13-oct-2016
> # ... AND uncommented the block below:
> #
27,28c36,37
< #userdb {
<   #driver = static
---
> userdb {
>   driver = static
30c39,43
< #}
---
>   # the following line came from:
>   # https://www.linode.com/docs/email/postfix/email-with-postfix-
dovecot-and-mysql
>   args = uid=vmail gid=vmail home=/var/mail/vhosts/%d/%n
> }
> #
```

O ficheiro "dovecot-sql.conf.ext" não existia. Foi criado da seguinte forma:

```
[root@machine2 dovecot]# vim dovecot-sql.conf.ext

[root@machine2 dovecot]# cat dovecot-sql.conf.ext
driver = mysql
connect = host=127.0.0.1 dbname=dbvmail user=vmail
password=[Password do utilizador]
default_pass_scheme = SHA512-CRYPT
password_query = SELECT email as user, password FROM virtual_users WHERE
email='%u';
```

```
[root@machine2 sysconfig]# cd /etc/pki/dovecot/certs

[root@machine2 certs]# cp -p dovecot.pem dovecot.pem.ORIG

[root@machine2 certs]# cp -p /etc/pki/tls/certs/www.uabdemo.info.crt .

[root@machine2 certs]# mv www.uabdemo.info.crt uabdemo.info.crt

[root@machine2 certs]# rm dovecot.pem
[root@machine2 certs]# ln -s uabdemo.info.crt dovecot.pem

[root@machine2 certs]# cd ../private/

[root@machine2 private]# cp -p dovecot.pem dovecot.pem.ORIG

[root@machine2 private]# cp -p /etc/pki/tls/private/www.uabdemo.info.key .
.

[root@machine2 private]# mv www.uabdemo.info.key uabdemo.info.key

[root@machine2 private]# rm dovecot.pem

[root@machine2 private]# ln -s uabdemo.info.key dovecot.pem

[root@machine2 private]# service dovecot restart

[root@machine2 var]# chown -R vmail:vmail mail/

[root@machine2 uabdemo.info]# mkdir ricardo.marques
[root@machine2 uabdemo.info]# chown -R vmail:vmail ricardo.marques/

[root@machine2 conf.d]# vim 10-ssl.conf
```

```
[root@machine2 conf.d]# cp -p 15-lda.conf 15-lda.conf.ORIG.20161022
[root@machine2 conf.d]# vim 15-lda.conf
[root@machine2 conf.d]# diff 15-lda.conf.ORIG.20161022 15-lda.conf
7a8,12
> # RICARDO
> # 22-out-2016
> # Adding the following line because I was having the following error
in "maillog":
> # dovecot: lmtp(2219): Fatal: Error reading configuration: Invalid
settings: postmaster_address setting not given
> postmaster_address = postmaster@uabdemo.info

[root@machine2 conf.d]# mv *.ORIG.* /root
[root@machine2 conf.d]# service dovecot restart
[root@machine2 etc]# chown -R vmail:dovecot /etc/dovecot/
[root@machine2 ~]# chkconfig dovecot on
```

10.3.7. Instalação e configuração do Roundcube

Para a instalação e configuração do Roundcube, foi consultada a documentação oficial localizada em: <https://github.com/roundcube/roundcubemail/wiki/Installation>

Os comandos usados para a instalação e configuração do Roundcube (e das suas dependências) foram os seguintes (foi omitido, no excerto abaixo, a *password* original):

```
[root@machine2 ~]# mkdir -p /root/installers/roundcube/
[root@machine2 ~]# cd /root/installers/roundcube/
[root@machine2 roundcube]# tar -xzvf roundcubemail-1.0.9.tar.gz
[root@machine2 roundcube]# cd roundcubemail-1.0.9/
[root@machine2 roundcubemail-1.0.9]# cd
/var/www/vhosts/www.uabdemo.info/
[root@machine2 webmail.uabdemo.info]# cp -pr
/root/installers/roundcube/roundcubemail-1.0.9 .
[root@machine2 webmail.uabdemo.info]# ln -s roundcubemail-1.0.9/
roundcubemail
```

```
[root@machine2 webmail.uabdemo.info]# mysql -u root -p
Enter password:
(...)

mysql> CREATE DATABASE roundcubemail /*!40101 CHARACTER SET utf8 COLLATE
utf8_general_ci */;

mysql> GRANT ALL PRIVILEGES ON roundcubemail.* TO roundcube@localhost
    IDENTIFIED BY '[Password do Utilizador]';

mysql> quit

[root@machine2 webmail.uabdemo.info]# cd roundcubemail

[root@machine2 roundcubemail]# mysql roundcubemail -u root -p <
SQL/mysql.initial.sql

[root@machine2 ~]# yum update

[root@machine2 ~]# yum install perl-xml

[root@machine2 ~]# yum install php-mbstring

[root@machine2 ~]# yum install libffi-devel

[root@machine2 ~]# yum install openssl-devel

[root@machine2 ~]# yum groupinstall 'development tools'

[root@machine2 ~]# yum install epel-release

[root@machine2 ~]# yum install php-mcrypt
Loaded plugins: fastestmirror
Setting up Install Process
Loading mirror speeds from cached hostfile
epel/metalink
(...)
epel
http://epel.check-update.co.uk/6/x86_64/repo/repodata/repomd.xml: [Errno -1]
repomd.xml does not match metalink for epel
Trying other mirror.

[root@machine2 yum.repos.d]# cp -p epel.repo epel.repo.ORIG.20161025

[root@machine2 yum.repos.d]# vim epel.repo
[root@machine2 yum.repos.d]# diff epel.repo.ORIG.20161025 epel.repo
3,4c3,10
< #baseurl=http://download.fedoraproject.org/pub/epel/6/$basearch
< mirrorlist=https://mirrors.fedoraproject.org/metalink?repo=epel-
6&arch=$basearch
---
```

```
> # RICARDO
> # 25-out-2016
> # Based on:
> # http://stackoverflow.com/questions/30949707/centos-6-6-errno-1-
repomd-xml-does-not-match-metalink-for-updates-when-tryin
> # ... I uncommented the following line:
> baseurl=http://download.fedoraproject.org/pub/epel/6/$basearch
> #... and commented the following line:
> #mirrorlist=https://mirrors.fedoraproject.org/metalink?repo=epel-
6&arch=$basearch
12,13c18,25
< #baseurl=http://download.fedoraproject.org/pub/epel/6/$basearch/debug
< mirrorlist=https://mirrors.fedoraproject.org/metalink?repo=epel-debug-
6&arch=$basearch
(...)

[root@machine2 ~]# yum install php-mcrypt
[root@machine2 ~]# yum install php-intl
[root@machine2 ~]# cd /etc
[root@machine2 etc]# cp -p php.ini php.ini.ORIG.20161025
[root@machine2 etc]# vim php.ini

[root@machine2 etc]# diff php.ini.ORIG.20161025 php.ini
946a947,950
> ; RICARDO
> ; 25-Oct-2016
> ; Added the following line for Roundcube installer:
> date.timezone = "UTC"

[root@machine2 webmail.uabdemo.info]# chown -R apache:apache
roundcubemail-1.0.9/

Adicionei a password do utilizador "roundcube" de Base de Dados em
"$config['db_dsnw'] = " no ficheiro
"/var/www/vhosts/webmail.uabdemo.info/roundcubemail/config"

[root@machine2 etc]# chown root:root postfix
[root@machine2 etc]# chmod 755 postfix/
```

10.3.8. Criação da Base de Dados de Mailings “dbmailings”

Os comandos usados para a criação da Base de Dados de Mailings “dbmailings” são os que constam da listagem abaixo. O conteúdo do ficheiro “dbmailings.sql” foi

listado anteriormente, no corpo deste Relatório. Foi omitida, na listagem abaixo, a *password* original:

```
[root@machine2 uabdemo.info]# mysql -u root -p
Enter password:
(...)
mysql> CREATE DATABASE dbmailings;

mysql> GRANT ALL PRIVILEGES ON dbmailings.* TO mailings_dbuser@localhost
      -> IDENTIFIED BY '[Password do Utilizador]';

mysql> GRANT ALL PRIVILEGES ON dbmailings.* TO mailings_dbuser@'%'
      IDENTIFIED BY '[Password do Utilizador]';

mysql> \. dbmailings.sql

mysql> FLUSH PRIVILEGES;

mysql> quit
```

10.3.9. Instalação de pacotes/módulos Perl

```
[root@machine2 ~]# yum install perl-Mail-IMAPClient
[root@machine2 ~]# yum install perl-MIME-EncWords
[root@machine2 ~]# yum install perl-List-MoreUtils
[root@machine2 ~]# yum install perl-Email-Address
[root@machine2 ~]# yum install perl-Email-MIME-Attachment-Stripper
```