



Instituto Superior de Contabilidade e Administração de Coimbra

# **Gestão de redes informáticas**

2.º Ano da Licenciatura em Informática de Gestão

**Ano Letivo  
2019-2020**



*Relatório do Trabalho Prático*

## **Configuração de Serviço de Correio Eletrónico**

Autores

**João Esculcas Nº 2018067460**

**André Vaz Nº 2018071303**

**Pedro Almeida Nº 2018069272**

Elaborado em  
**Julho de 2020**



## Índice

Introdução.....	3
Estudo dos conceitos fundamentais do serviço.....	4
<b>Correio Eletrónico</b> .....	4
<b>Protocolo SMTP</b> .....	5
<b>Protocolo POP</b> .....	5
<b>Protocolo IMAP</b> .....	6
<b>Servidor SMTP, DNS e MTA</b> .....	6
<b>Sendmail</b> .....	7
<b>Dovecot</b> .....	7
Cenário Prático .....	8
Configuração do servidor de POP e IMAP - Dovecot.....	9
Configuração do servidor SMTP Sendmail .....	15
<b>Aceitar e-mail de outros domínios</b> .....	18
<b>Retransmissão de E-Mails com sendmail</b> .....	20
<b>Assunto e Mensagem numa única linha</b> .....	22
<b>Verbose output</b> .....	22
<b>Blackhole list</b> .....	23
<b>Smart Host</b> .....	24
<b>Make Auth Directory</b> .....	25
<b>Masquerading</b> .....	27
<b>Aliases – envio de mail local</b> .....	28
Configuração do cliente de correio eletrónico .....	31
Conclusão.....	35



## Introdução

No âmbito da unidade curricular de Gestão de redes informáticas, foi-nos proposto como projeto de trabalho a implementação de um serviço de correio eletrónico, a desenvolver no sistema operativo “Linux”, no nosso caso utilizando a distribuição Ubuntu, e a configuração de um dispositivo cliente de serviço de correio eletrónico. Para o sucesso do projeto em questão, é necessário configurar e implementar os seguintes servidores: SMTP (Simple Mail Transfer Protocol), POP (Post Office Protocol) e o IMAP (Internet Message Access Protocol).

Numa fase inicial, e para que melhor possamos compreender todos os conceitos necessários, elaborámos uma abordagem teórica para melhor aprofundar os conhecimentos do leitor.



## Estudo dos conceitos fundamentais do serviço

### Correio Eletrónico

De forma a compreendermos os conceitos sobre este trabalho, necessitamos compreender o que é um serviço de correio eletrónico.

Um serviço de correio eletrónico consiste num serviço que permite a troca de mensagens através de sistemas de comunicação eletrónicos. As mensagens de correio eletrónico possibilitam o envio, não só de textos, como de qualquer tipo de documento digital.

Podemos seguir a história do correio eletrónico até ao final da década de 60 e início da década de 70. Nessa altura já existiam sistemas de troca de mensagens, no entanto a comunicação era apenas possível entre vários utilizadores de um computador *mainframe*.

De forma a que os serviços de correio eletrónico pudessem evoluir, foi necessário a utilização da rede desenvolvida pelo ARPANET. Com esta foi possível criar os sistemas que hoje conhecemos.

Com a evolução foi preciso criar uma forma única endereço de correio eletrónico. Este endereço é ainda usado nos dias de hoje e é composto da seguinte forma, [nomeUtilizador@dominio.ext](#), onde o “nomeUtilizador” é único para cada um e o “dominio” depende do servidor que esta a prestar serviço.

Com o aumento de domínios, foi necessário introduzir algo que diferenciase o utilizador do domínio. Foi então que o programador Ray Tomlinson, em 1971, introduziu o carácter @ para separar os nomes de utilizador do correspondente domínio (servidor).



## Protocolo SMTP

O protocolo SMTP (Simple Mail Transfer Protocol) teve o início da sua utilização em massa nos anos 80, como forma de complemento ao programa UUCP, para a transferência de correio eletrónico entre máquinas sem ligação permanente. Atualmente, o protocolo SMTP continua a ser bastante utilizado, uma vez que é o protocolo padrão de envio de mensagens de correio eletrónico entre duas máquinas através da internet.

Este protocolo tem a função somente de envio, isto é, recebe a mensagem do usuário onde é especificado o/os destinatários e transfere a mesma para o servidor respetivo ao destinatário, normalmente através da porta TCP 25, havendo, claro, exceções.

## Protocolo POP

O protocolo POP (Post Office Protocol), também referido como POP3, é um protocolo utilizado no acesso remoto a um servidor de correio eletrónico que permite a transferência de mensagens para um software cliente noutro dispositivo. Este protocolo serve-se mais frequentemente da porta TCP 110, anteriormente porta TCP 109, aquando da sua versão POP2.

Este protocolo atravessa 5 etapas no seu processo de funcionamento:

- Estabelece-se a relação TCP entre a aplicação de e-mail cliente e o servidor de correio eletrónico onde se encontra a caixa de entrada
- O usuário autentica-se
- As mensagens em caixa de entrada são automaticamente transferidas para o computador local
- A caixa de correio fica vazia após a eliminação das respetivas mensagens
- A ligação com o utilizador é terminada e o utilizador pode ler as suas mensagens off-line



## Protocolo IMAP

O protocolo IMAP (Internet Message Access Protocol) consiste num protocolo de gestão de correio eletrónico semelhante ao POP, no entanto mais avançado. A principal diferença entre os dois é que o protocolo IMAP guarda uma cópia da mensagem no servidor, permitindo desta forma que o utilizador aceda ao seu correio em qualquer dispositivo, desde que possua uma conexão à internet, simultânea ou individualmente.

Este protocolo serve-se da porta TCP 143, salvo exceções para conexões encriptadas, para as quais se usa a porta 993.

## Servidor SMTP, DNS e MTA

O protocolo SMTP (Simple Mail Transfer Protocol) é um dos principais responsáveis por todo o processo de entrega de um e-mail. Neste processo, o servidor SMTP consiste num computador que recebe as mensagens enviadas e se responsabiliza de enviá-las para os respetivos destinatários.

Para conseguir remeter o correio eletrónico, não basta ao protocolo SMTP saber o domínio, uma vez que não o consegue decifrar. Como solução a este problema, o protocolo SMTP contacta um servidor DNS (Domain Name System) que lhe traduzirá os domínios para endereços IP.

Chegado a este ponto, o protocolo SMTP é finalmente capaz de enviar a mensagem para o servidor de troca de correio do domínio em questão. A este servidor damos o nome de MTA (Mail Transfer Agent), servidor responsável pelo envio do e-mail diretamente para o cliente, servindo-se a arquitetura de aplicações cliente-servidor. No contexto do nosso projeto de trabalho, utilizaremos o MTA sendmail, como recomendado pela docente da unidade curricular.



## Sendmail

O Sendmail consiste num agente de transferência de correio, ou MTA (Mail Transfer Agent) como abordado anteriormente, que através de diferentes métodos e servindo-se de diferentes protocolos, nomeadamente o SMTP, se encarrega de transferir mensagens de e-mail de uma máquina para a outra, através da sua arquitetura cliente-servidor

Lançado em 1983 e programado através das linguagens Perl e C, este agente de transferência de correio é um descendente direto do Delivermail, escrito pelo mesmo programador, Eric Allman.

## Dovecot

O Dovecot é um servidor de IMAP e POP3 open source para sistemas Linux e Unix. O principal objetivo deste servidor é ser um servidor de email open source rápido, leve e de fácil configuração. Este servidor inclui ainda um agente de entrega de emails chamado Local Delivery Agent

Lançado em Julho de 2002 e programado na linguagem C, este servidor de e-mail foi escrito pelo programador Timo Sirainen, e já vai na versão 2.3.9.2, tendo sido atualizado a última vez a 13 de Dezembro de 2019.



## Cenário Prático

Linux é o Sistema Operativo sendo o ubuntu a distribuição utilizada.

Recorremos ao ficheiro hosts, de forma a definir os servidores smtp.gri.pt e mail.gri.pt como hosts, atribuindo-lhes também o respetivo IP, que será o IP da máquina em questão, que obtivemos através do comando ifconfig.

```
gedit /etc/hosts
```

```
10.0.2.15 smtp.gri.pt mail.gri.pt
```

Antes de começar, é necessário salientar que foi preciso ter acesso a uma conta com permissões de root, ou então utilizar no início de cada comando a aplicação SUDO, o que se torna mais trabalhoso e propenso a erro/esquecimento, sendo que optámos pela primeira opção.

Para testar as configurações iremos criar vários utilizadores, ao qual vamos atribuir os nomes e as respetivas palavras passe.

```
Adduser teste1
```

```
Adduser teste2
```

```
Adduser teste3
```

```
Adduser teste4
```

```
root@lordep:/home# adduser teste1
Adding user `teste1' ...
Adding new group `teste1' (1001) ...
Adding new user `teste1' (1001) with group `teste1' ...
Creating home directory `/home/teste1' ...
Copying files from `/etc/skel' ...
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
Changing the user information for teste1
Enter the new value, or press ENTER for the default
  Full Name []:
  Room Number []:
  Work Phone []:
  Home Phone []:
  Other []:
Is the information correct? [Y/n] y
root@lordep:/home# adduser teste2
Adding user `teste2' ...
Adding new group `teste2' (1002) ...
Adding new user `teste2' (1002) with group `teste2' ...
Creating home directory `/home/teste2' ...
Copying files from `/etc/skel' ...
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
Changing the user information for teste2
Enter the new value, or press ENTER for the default
  Full Name []:
  Room Number []:
  Work Phone []:
  Home Phone []:
  Other []:
Is the information correct? [Y/n] y
root@lordep:/home#
```





## Configuração do servidor de POP e IMAP - Dovecot

Executámos o comando update para atualizar os repositórios e obter as informações mais recentes sobre os pacotes:

```
apt-get update -y
```

'y' significa assumir sim e instalar silenciosamente, sem fazer perguntas na maioria dos casos.

Verificar se o serviço já se encontrava na máquina onde foi executado o comando:

```
systemctl status dovecot
```

Como tal não se verificava, instalar o serviço:

```
apt-get install dovecot-imapd dovecot-pop3d
```

Verificação de inicialização do serviço:

```
systemctl status dovecot
```

```
root@lordep: /home/lordep
dovecot.service - Dovecot IMAP/POP3 email server
Loaded: loaded (/lib/systemd/system/dovecot.service; enabled; vendor preset: enabled)
Active: active (running) since Thu 2020-07-16 00:35:34 WEST; 4min 29s ago
Docs: man:dovecot(1)
      http://wiki2.dovecot.org/
Main PID: 819 (dovecot)
Tasks: 4 (limit: 2328)
CGroup: /system.slice/dovecot.service
        └─819 /usr/sbin/dovecot -F
          └─943 dovecot/anvil
            └─944 dovecot/log
              └─951 dovecot/config

jul 16 00:35:34 lordep systemd[1]: Started Dovecot IMAP/POP3 email server.
jul 16 00:35:35 lordep dovecot[819]: master: Dovecot v2.2.33.2 (d6601f4ec) starting up for imap, pop3 (core dumps disabled)
```

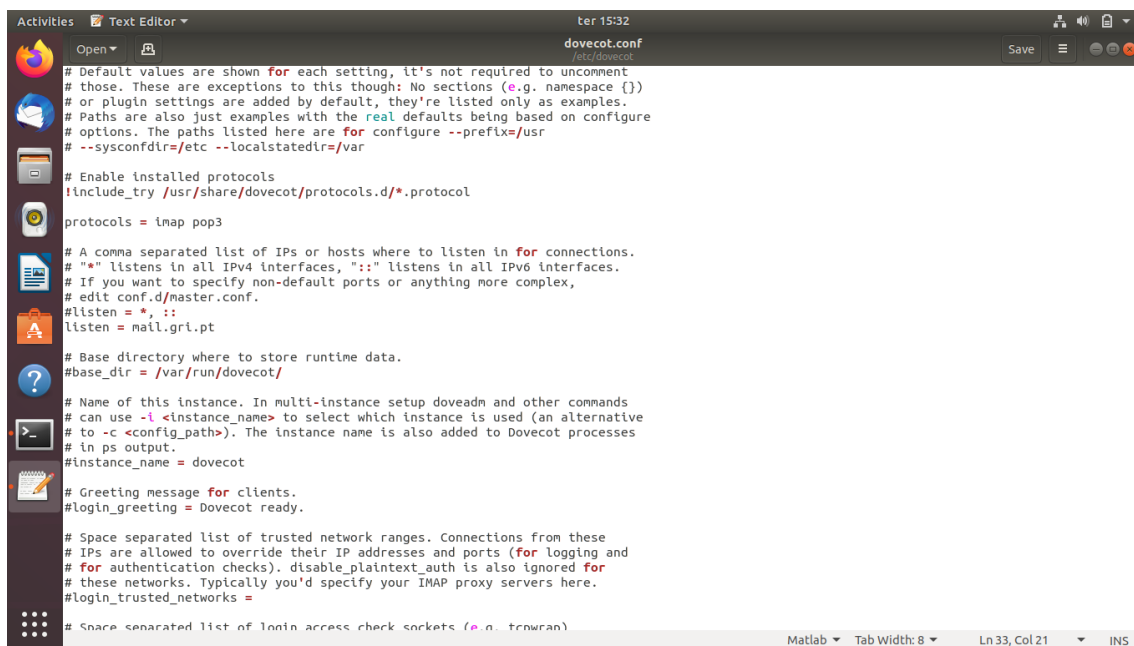


Depois de instalado vamos editar o ficheiro dovecot.conf, para especificarmos os protocolos que queremos (imap e o pop3) e colocar o domínio mail.gri.pt em escuta.

```
gedit /etc/dovecot.conf
```

```
protocols = imap pop3
```

```
listen = mail.gri.pt
```



```
# Default values are shown for each setting, it's not required to uncomment
# those. These are exceptions to this though: No sections (e.g. namespace {})
# or plugin settings are added by default, they're listed only as examples.
# Paths are also just examples with the real defaults being based on configure
# options. The paths listed here are for configure --prefix=/usr
# --sysconfdir=/etc --localstatedir=/var

# Enable installed protocols
!include_try /usr/share/dovecot/protocols.d/*.protocol

protocols = imap pop3

# A comma separated list of IPs or hosts where to listen in for connections.
# "*" listens in all IPv4 interfaces, "::" listens in all IPv6 interfaces.
# If you want to specify non-default ports or anything more complex,
# edit conf.d/master.conf.
#listen = *, ::
listen = mail.gri.pt

# Base directory where to store runtime data.
#base_dir = /var/run/dovecot/

# Name of this instance. In multi-instance setup doveadm and other commands
# can use -i <instance_name> to select which instance is used (an alternative
# to -c <config_path>). The instance name is also added to Dovecot processes
# in ps output.
#instance_name = dovecot

# Greeting message for clients.
#login_greeting = Dovecot ready.

# Space separated list of trusted network ranges. Connections from these
# IPs are allowed to override their IP addresses and ports (for logging and
# for authentication checks). disable_plaintext_auth is also ignored for
# these networks. Typically you'd specify your IMAP proxy servers here.
#login_trusted_networks =

# Space separated list of login access check sockets (e.g. tcpwrap)
```

Posteriormente, foi necessário alterar as configurações de outros três ficheiros que estavam na diretoria dovecot na pasta conf.d: 10-auth.conf, 10-mail.conf, 10-master.conf.

```
gedit /etc/dovecot/conf.d/10-auth.conf
```

```
disable_plaintext_auth = yes
```

```
auth_mechanisms = plain login
```



## Instituto Superior de Contabilidade e Administração de Coimbra

```
Activities Text Editor ter 15:36
*10-auth.conf
/etc/dovecot/conf.d

## Authentication processes
##

# Disable LOGIN command and all other plaintext authentications unless
# SSL/TLS is used (LOGINDISABLED capability). Note that if the remote IP
# matches the local IP (ie. you're connecting from the same computer), the
# connection is considered secure and plaintext authentication is allowed.
# See also ssl=required setting.
disable_plaintext_auth = yes

# Authentication cache size (e.g. 10M). 0 means it's disabled. Note that
# bsdauth, PAM and vpopmail require cache_key to be set for caching to be used.
#auth_cache_size = 0
# Time to live for cached data. After TTL expires the cached record is no
# longer used, except if the main database lookup returns internal failure.
# We also try to handle password changes automatically: If user's previous
# authentication was successful, but this one wasn't, the cache isn't used.
# For now this works only with plaintext authentication.
#auth_cache_ttl = 1 hour
# TTL for negative hits (user not found, password mismatch).
# 0 disables caching them completely.
#auth_cache_negative_ttl = 1 hour

# Space separated list of realms for SASL authentication mechanisms that need
# them. You can leave it empty if you don't want to support multiple realms.
# Many clients simply use the first one listed here, so keep the default realm
# first.
#auth_realms =

# Default realm/domain to use if none was specified. This is used for both
# SASL realms and appending @domain to username in plaintext logins.
#auth_default_realm =

# List of allowed characters in username. If the user-given username contains
# a character not listed in here, the login automatically fails. This is just
# an extra check to make sure user can't exploit any potential quote escaping
# vulnerabilities with SQL/LDAP databases. If you want to allow all characters

Take the username from client's SSL certificate, using
# X509_NAME_get_text_by_NID() which returns the subject's DN's
# CommonName.
#auth_ssl_username_from_cert = no

# Space separated list of wanted authentication mechanisms:
# plain login digest-md5 cram-md5 ntlm rpa apop anonymous gssapi otp skey
# gss-spnego
# NOTE: See also disable_plaintext_auth setting.
auth_mechanisms = plain login

## Password and user databases
##

# Password database is used to verify user's password (and nothing more).
# You can have multiple passwd and userdb. This is useful if you want to
# allow both system users (/etc/passwd) and virtual users to login without
# duplicating the system users into virtual database.
#
# <doc/wiki/PasswordDatabase.txt>
#
# User database specifies where mails are located and what user/group IDs
# own them. For single-UID configuration use "static" userdb.
#
# <doc/wiki/UserDatabase.txt>
#!include auth-deny.conf.ext
#!include auth-master.conf.ext

!include auth-system.conf.ext
#!include auth-sql.conf.ext
#!include auth-ldap.conf.ext
#!include auth-passwdfile.conf.ext
#!include auth-checkpassword.conf.ext
#!include auth-vpopmail.conf.ext
#!include auth-static.conf.ext

Matlab Tab Width: 8 Ln 100, Col 30 INS
```

```
Activities Text Editor ter 15:36
*10-auth.conf
/etc/dovecot/conf.d

Take the username from client's SSL certificate, using
# X509_NAME_get_text_by_NID() which returns the subject's DN's
# CommonName.
#auth_ssl_username_from_cert = no

# Space separated list of wanted authentication mechanisms:
# plain login digest-md5 cram-md5 ntlm rpa apop anonymous gssapi otp skey
# gss-spnego
# NOTE: See also disable_plaintext_auth setting.
auth_mechanisms = plain login

## Password and user databases
##

# Password database is used to verify user's password (and nothing more).
# You can have multiple passwd and userdb. This is useful if you want to
# allow both system users (/etc/passwd) and virtual users to login without
# duplicating the system users into virtual database.
#
# <doc/wiki/PasswordDatabase.txt>
#
# User database specifies where mails are located and what user/group IDs
# own them. For single-UID configuration use "static" userdb.
#
# <doc/wiki/UserDatabase.txt>
#!include auth-deny.conf.ext
#!include auth-master.conf.ext

!include auth-system.conf.ext
#!include auth-sql.conf.ext
#!include auth-ldap.conf.ext
#!include auth-passwdfile.conf.ext
#!include auth-checkpassword.conf.ext
#!include auth-vpopmail.conf.ext
#!include auth-static.conf.ext

Matlab Tab Width: 8 Ln 100, Col 30 INS
```

A seguir acedemos ao ficheiro 10-mail.conf e removemos o comentário na linha a especificar a localização do email.

```
gedit /etc/dovecot/conf.d/10-mail.conf
```

```
mail_location = mbox:~/mail:INBOX~/var/mail/%u
```



## Instituto Superior de Contabilidade e Administração de Coimbra

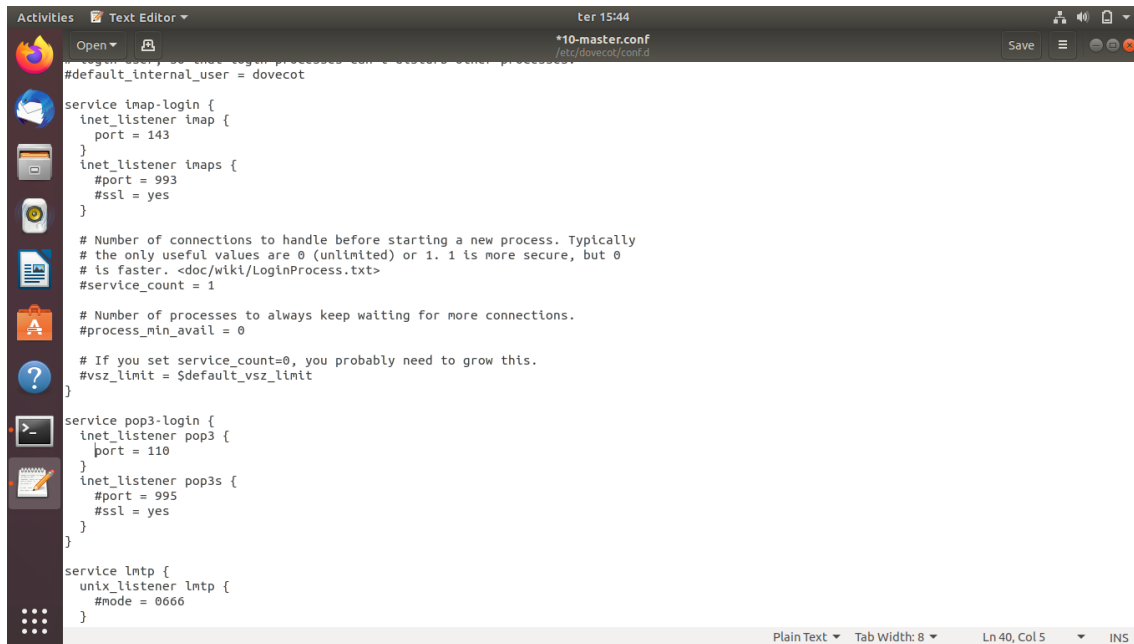
```
# path given in the mail_location setting.
#
# There are a few special variables you can use, eg.:
#
# %u - username
# %n - user part in user@domain, same as %u if there's no domain
# %d - domain part in user@domain, empty if there's no domain
# %h - home directory
#
# See doc/wiki/Variables.txt for full list. Some examples:
#
# mail_location = maildir:~/Maildir
# mail_location = mbox:~/mail:INBOX=/var/mail/%u
# mail_location = mbox:/var/mail/%d/%ln/%n:INDEX=/var/indexes/%d/%ln/%n
#
# <doc/wiki/MailLocation.txt>
#
mail_location = mbox:~/mail:INBOX=/var/mail/%u
#
# If you need to set multiple mailbox locations or want to change default
# namespace settings, you can do it by defining namespace sections.
#
# You can have private, shared and public namespaces. Private namespaces
# are for user's personal mails. Shared namespaces are for accessing other
# users' mailboxes that have been shared. Public namespaces are for shared
# mailboxes that are managed by sysadmin. If you create any shared or public
# namespaces you'll typically want to enable ACL plugin also, otherwise all
# users can access all the shared mailboxes, assuming they have permissions
# on filesystem level to do so.
namespace inbox {
# Namespace type: private, shared or public
#type = private
#
# Hierarchy separator to use. You should use the same separator for all
# namespaces or some clients get confused. '/' is usually a good one.
# The default however depends on the underlying mail storage format.
#separator =
```

Por último na diretoria conf.d, acedemos ao ficheiro 10-master.conf onde iremos tirar os comentários relativos às portas dos protocolos (imap e pop3) e ceder permissões ao sendmail.

```
gedit /etc/dovecot/conf.d/10-master.conf
```

```
service imap-login {
    inet_listener imap {
        port = 143
    }
}

service pop3-login {
    inet_listener pop3 {
        port = 110
    }
}
```



```
#default_internal_user = dovecot

service imap-login {
  inet_listener imap {
    port = 143
  }
  inet_listener imaps {
    #port = 993
    #ssl = yes
  }
}

# Number of connections to handle before starting a new process. Typically
# the only useful values are 0 (unlimited) or 1. 1 is more secure, but 0
# is faster. <doc/wikit/LoginProcess.txt>
#service_count = 1

# Number of processes to always keep waiting for more connections.
#process_min_avail = 0

# If you set service_count=0, you probably need to grow this.
#vsz_limit = $default_vsz_limit
}

service pop3-login {
  inet_listener pop3 {
    port = 110
  }
  inet_listener pop3s {
    #port = 995
    #ssl = yes
  }
}

service lntp {
  unix_listener lntp {
    #mode = 0666
  }
}
```

Feitas estas modificações nos três ficheiros, reiniciámos o Dovecot.

```
service dovecot restart
```

Verificámos o estado das ligações de rede ao servidor, isto é, se os portos estavam à escuta:

```
Netstat -nat | grep 110
```

```
Netstat -nat | grep 143
```

E testámos o sucesso da configuração do dovecot com a tentativa de conexão ao localhost pelos portos do POP3 e do IMAP, respetivamente:

```
telnet localhost 110
```

```
telnet localhost 143
```



## Instituto Superior de Contabilidade e Administração de Coimbra

```
Activities Terminal ter 16:17
root@lordep:/etc/dovecot/conf.d
File Edit View Search Terminal Help
root@lordep:/etc/dovecot/conf.d# netstat -nat | grep 110
tcp        0      0 0.0.0.0:110 0.0.0.0:*      LISTEN
tcp6       0      0 :::110     :::*           LISTEN
root@lordep:/etc/dovecot/conf.d# netstat -nat | grep 143
tcp        0      0 0.0.0.0:143 0.0.0.0:*      LISTEN
tcp6       0      0 :::143     :::*           LISTEN
root@lordep:/etc/dovecot/conf.d# telnet localhost 110
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
+OK Dovecot (Ubuntu) ready.
quit
+OK Logging out
Connection closed by foreign host.
root@lordep:/etc/dovecot/conf.d# telnet localhost 143
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
* OK [CAPABILITY IMAP4rev1 LITERAL+ SASL-IR LOGIN-REFERRALS ID ENABLE IDLE STARTTLS AUTH=PLAIN] Dovecot (Ubuntu) ready.
all logout
* BYE logging out
all OK Logout completed.
Connection closed by foreign host.
root@lordep:/etc/dovecot/conf.d#
```



## Configuração do servidor SMTP Sendmail

Verificar se o serviço já se encontrava na máquina onde foi executado o comando:

```
systemctl status sendmail
```

Como tal não se verificava, instalar o serviço:

```
apt-get install sendmail
```

Instalar m4 tools:

```
apt-get install -y m4
```

Verificar a diretoria principal:

```
ls -l /etc/mail
```

```
root@lordep:/etc/dovecot/conf.d# ls -l /etc/mail
total 220
-rw-r----- 1 root root 4311 jun 26 23:59 access
-rw-r----- 1 smmta smmsp 12288 jun 26 23:59 access.db
-rw-r----- 1 root root 281 jan 13 2018 address.resolve
lrwxrwxrwx 1 root smmsp 10 jun 26 16:28 aliases -> ../aliases
-rw-r----- 1 smmta smmsp 12288 jun 27 00:06 aliases.db
-rw-r----- 1 root smmsp 3223 jun 27 01:06 databases
-rw-r----- 1 root root 5659 jan 13 2018 helpfile
-rw-r----- 1 root smmsp 24 jun 26 23:32 local-host-names
drwxr-sr-x 2 smmta smmsp 4096 jun 26 16:28 m4
-rwxr-xr-x 1 root smmsp 10022 jun 27 01:06 Makefile
drwxr-xr-x 2 root root 4096 jun 26 16:28 peers
drwxr-xr-x 2 root smmsp 4096 jan 13 2018 sasl
-rw-r----- 1 root smmsp 60539 jun 27 01:06 sendmail.cf
-rw-r----- 1 root root 12235 jun 26 16:28 sendmail.conf
-rw-r----- 1 root smmsp 4245 jun 27 01:06 sendmail.mc
-rw-r----- 1 root root 148 jan 13 2018 service.switch
-rw-r----- 1 root root 179 jan 13 2018 service.switch-nodns
drwxr-sr-x 2 smmta smmsp 4096 jun 26 16:28 smrsh
-rw-r----- 1 root smmsp 44002 jun 26 16:28 submit.cf
-rw-r----- 1 root smmsp 2375 jun 26 16:28 submit.mc
drwxr-xr-x 2 smmta smmsp 4096 jun 26 16:28 tls
-rw-r----- 1 root smmsp 0 jun 26 16:28 trusted-users
root@lordep:/etc/dovecot/conf.d#
```



## Verificação de inicialização do serviço:

Systemctl status sendmail

```
Activities Terminal qua 15:03
root@lordep: /etc/dovecot/conf.d

File Edit View Search Terminal Help

root@lordep:/etc/dovecot/conf.d# systemctl status sendmail
● sendmail.service - LSB: powerful, efficient, and scalable Mail Transport Agent
   Loaded: loaded (/etc/init.d/sendmail; generated)
   Active: active (running) since Wed 2020-07-15 13:11:47 WEST; 1h 51min ago
     Docs: man:systemd-sysv-generator(8)
   Process: 809 ExecStart=/etc/init.d/sendmail start (code=exited, status=0/SUCCESS)
    Tasks: 1 (limit: 2328)
   CGroup: /system.slice/sendmail.service
           └─1577 sendmail: MTA: accepting connections

jul 15 13:10:38 lordep su[894]: Successful su for smmsp by root
jul 15 13:10:38 lordep su[894]: + ??? root:smmsp
jul 15 13:10:38 lordep su[894]: pam_unix(su:session): session opened for user smmsp by (uid=0)
jul 15 13:10:42 lordep sm-nta[1010]: My unqualified host name (lordep) unknown; sleeping for retry
jul 15 13:10:43 lordep sm-nsp-queue[1046]: My unqualified host name (lordep) unknown; sleeping for retry
jul 15 13:11:46 lordep sm-nta[1010]: unable to qualify my own domain name (lordep) -- using short name
jul 15 13:11:46 lordep sm-nsp-queue[1046]: starting daemon (8.15.2): SMTP+queueing@00:10:00
jul 15 13:11:47 lordep sm-nsp-queue[1046]: unable to qualify my own domain name (lordep) -- using short name
jul 15 13:11:47 lordep sendmail[809]: ...done.
jul 15 13:11:47 lordep systemd[1]: Started LSB: powerful, efficient, and scalable Mail Transport Agent.
root@lordep:/etc/dovecot/conf.d#
```

## Verificámos se o porto 25 estava à escuta:

netstat -nat | grep 25

```
Activities Terminal qua 15:16
root@lordep: /etc/dovecot/conf.d

File Edit View Search Terminal Help

root@lordep:/etc/dovecot/conf.d# netstat -nat | grep 25
tcp        0      0 0.0.0.0:25 0.0.0.0:*        LISTEN
root@lordep:/etc/dovecot/conf.d#
```

À escuta apenas do localhost.





Tentativa de conexão com o localhost pela porta 25:

```
telnet localhost 25
```

```
root@lordep:/etc/dovecot/conf.d# telnet localhost 25
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
220 lordep ESMTTP Sendmail 8.15.2/8.15.2/Debian-10; Wed, 15 Jul 2020 15:06:56 +0100; (No UCE/UBE) logging access from: localhost(OK)-localhost
[127.0.0.1]
quit
221 2.0.0 lordep closing connection
Connection closed by foreign host.
root@lordep:/etc/dovecot/conf.d#
```

## Configuração do e-mail de entrada

Por padrão, o Sendmail permitirá apenas entrada de e-mail do localhost. De forma a contrariar tal acontecimento alterámos a seguinte linha no ficheiro sendmail.mc:

```
gedit /etc/mail/sendmail.mc
```

```
DAEMON_OPTIONS(`Family=inet, Name=MTA-v4, Port=smtp,Addr=127.0.0.1')dnl
```

```
Para: DAEMON_OPTIONS(`Family=inet, Name=MTA-v4, Port=smtp')dnl
```

Para garantir que o sendmail ouça na porta 25:

```
service sendmail restart
```

```
netstat -nat | grep 25
```



```
root@lordep:/etc/dovecot/conf.d# netstat -nat | grep 25
tcp        0      0 0.0.0.0:25 0.0.0.0:*    LISTEN
root@lordep:/etc/dovecot/conf.d#
```

Os pedidos na porta 25 estão sendo atendidos de todas as direções.

## Aceitar e-mail de outros domínios

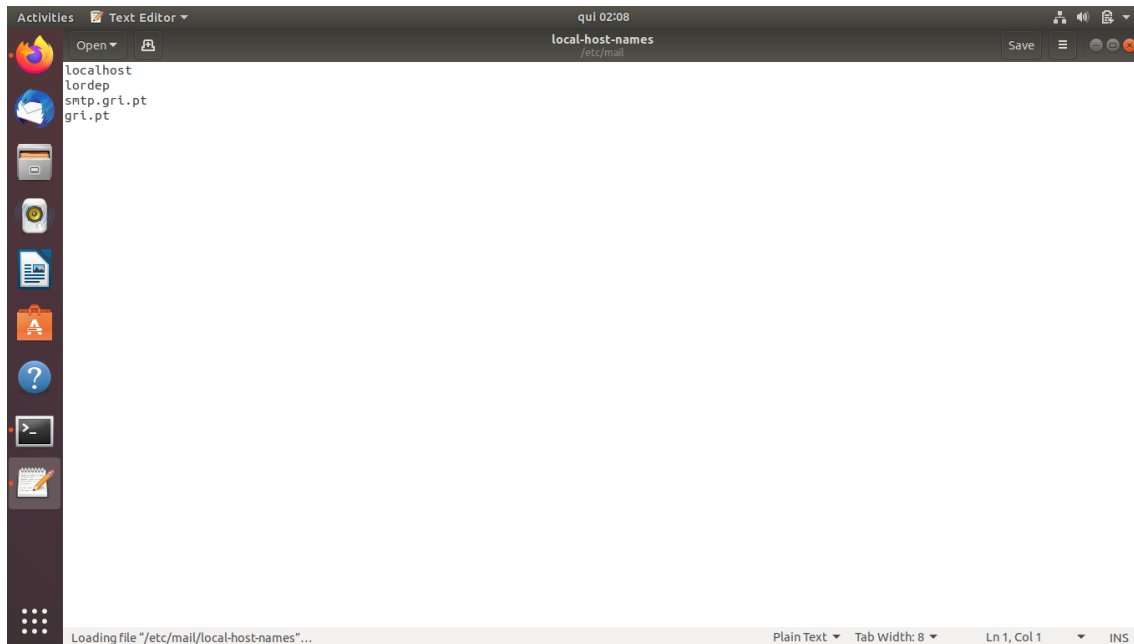
O Sendmail não aceita e-mails de um certo domínio/host, a menos que este seja permitido especificamente no ficheiro local-host-names

Assim, neste arquivo adicionámos o domínio/host pelo qual o sendmail irá aceitar a entrega de correio eletrónico.

```
gedit /etc/mail/local-host-names
```

```
smtp.gri.pt
```

```
gri.pt
```



Quando o Sendmail recebe uma mensagem de e-mail, compara o host de destino no cabeçalho da mensagem com o nome do host presente neste ficheiro. Se ambos coincidirem, o Sendmail aceita a mensagem para entrega local.

A feature `use_cw_file` habilita o ficheiro onde estão guardados os domínios para o qual o sendmail aceita correio. Para a configurar, basta adicionar a seguinte declaração no ficheiro `sendmail.mc`:

```
FEATURE(use_cw_file)
```

Como esta já vinha implementada por defeito, tal não foi necessário.

## Configuração do envio de e-mail

Por padrão, o Sendmail permitirá apenas retransmissão de e-mail pelo localhost. Para contrariar tal acontecimento a seguinte alteração, no ficheiro `sendmail.mc`, era necessária:

```
gedit /etc/mail/sendmail.mc
```

```
DAEMON_OPTIONS(`Port=smtp,Addr=127.0.0.1, Name=MTA')
```

Para: `DAEMON_OPTIONS(`Port=smtp, Name=MTA')`

Alteração esta já efetuada anteriormente para a entrada de e-mail.



## Retransmissão de E-Mails com sendmail

A retransmissão e o recebimento de emails são controlados pelo ficheiro access.

Este arquivo decide de quais os sites ou servidores é permitido receber correio eletrónico e quais são recusados.

A ideia geral é simples. Quando uma nova conexão SMTP é recebida, o sendmail recupera as informações do cabeçalho da mensagem e consulta o banco de dados de acesso para ver se deve continuar e, aceitar o corpo da mensagem.

A base de dados de acesso é um conjunto de regras que descrevem quais ações devem ser tomadas para mensagens recebidas de hosts nomeados. É útil para bloquear alguns emails indesejados, ou seja, serve para rejeitar todos os emails originários de spammers conhecidos.

Cada rede ou host interno para o qual o servidor retransmitirá e rejeitará e-mail:

```
gedit /etc/mail/access
```

```
Connect: 10.0.2.15      RELAY
Connect: gri.pt         RELAY
Connect: spam.net       REJECT
```

```
Activities Text Editor qua 16:49
access
/etc/mail
Save
GreetPause: 5000
ClientRate: 10
ClientConn: 10
#
# Don't offer AUTH on local network
#SRV_Features:192.168.1 A
#
# Hosts with to allow relaying
Connect:10.0.2.15 RELAY
Connect:gri.pt RELAY
#
# Hosts that validly forward to me
#GreetPause:<ip> 0
#ClientRate:<ip> 30
#ClientConn:<ip> 0
#
# Whitelisted users
#
Spam:postmaster@ FRIEND
Spam:abuse@ FRIEND
Spam:spam@ FRIEND
#
# Blacklisted users
#
#Connect:rampellsoft.com 554 Email directly, not through didtheyreadit.com
reject@ REJECT
#Cyberpromo.com REJECT
#From:MAILER-DAEMON@store2.netvisao.pt REJECT
#
# Block invalid IPs
#
#Connect:0 REJECT whilst invalid, this also blocks sendmail -bs -Am
Connect:169.254 REJECT
Connect:192.0.2 REJECT
Connect:224 REJECT
Connect:255 REJECT
Connect:spam.net REJECT
```



A tabela possui um formato simples. O lado esquerdo (por exemplo, "Connect:spam.net") é um padrão usado para corresponder ao remetente de uma mensagem de email recebida. O lado direito é a ação a ser tomada ("REJECT"). Existem cinco tipos de ações possíveis de configuração. Estas são:

**OK** - Aceitar correio, mesmo que outras regras do conjunto de regras em execução o rejeitem, por exemplo, se o nome do domínio não se encontre definido no ficheiro local-host-names

**RELAY** - Aceita correio do domínio indicado para a retransmissão. Como um OK implícito para as outras verificações.

**REJECT** - Rejeitar o remetente ou destinatário com uma mensagem de uso geral.

**DISCARD** – Descarta completamente a mensagem usando o discard mailer. Funciona apenas para endereços de remetente.

**### texto** - Retorna uma mensagem de erro usando ### como o código de erro e "texto" como a mensagem.

Como /etc/mail/access é uma base de dados, depois de criar o arquivo de texto conforme descrito abaixo, deve-se usar o makemap para criar o mapa da base de dados.

```
makemap hash /etc/mail/access.db < /etc/mail/access
```

Para ativar a feature access.db, a seguinte declaração é necessária no ficheiro sendmail.mc:

```
FEATURE(access_db)
```

Como esta já vinha implementada por defeito, não foi necessário implementar.

Recorremos a macros M4Tools, sendo estas responsáveis pela criação do ficheiro de configuração principal do serviço:

```
m4 /etc/mail/sendmail.mc > /etc/mail/sendmail.cf
```



Reiniciámos o serviço:

```
service sendmail restart
```

## Teste à entrada e saída de e-mail

Primeiramente, uma breve descrição sobre as formas de envio de email com comandos.

### Assunto e Mensagem numa única linha

Para especificar o corpo da mensagem numa única linha de comando, use a seguinte linha de comandos

```
mail -s "This is the subject" somebody@example.com <<< 'This is the message'
```

ou, então:

```
echo "This is the message body" | mail -s "This is the subject" mail@example.com
```

### Verbose output

Por vezes, ao testar servidores de e-mail, é necessário testar os comandos SMTP utilizados pelo comando mail. Para isso, utilizamos a opção “-v”

```
mail -v -s "This is the subject" somebody@example.com <<< 'This is the message'
```

Caso haja uma falha na entrega devido a uma configuração imprópria do servidor de e-mail, por exemplo, o comando SMTP indicará o que está mal.

No nosso caso, infelizmente, esta opção não se encontrava disponível.



## Teste:

```
Activities Terminal qua 16:45
teste1@lordep: ~
File Edit View Search Terminal Help
root@lordep:/home# mail -s "envio e entrada de e-mail" teste1@gri.pt
Cc:
Este é um e-mail de teste ao envio e entrada de correio eletrónico de root para teste1
root@lordep:/home# su - teste1
teste1@lordep:~$ mail
"/var/mail/teste1": 1 message 1 new
>N 1 root qua jul 15 16:44 15/603 envio e entrada de e-mail
? 1
Return-Path: <root@lordep>
Received: from lordep (localhost [127.0.0.1])
by lordep (8.15.2/8.15.2/Debian-10) with ESMTP id 06FFICKV003821
for <teste1@gri.pt>; Wed, 15 Jul 2020 16:44:12 +0100
Received: (from root@localhost)
by lordep (8.15.2/8.15.2/Submit) id 06FFICD0003815;
Wed, 15 Jul 2020 16:44:12 +0100
Date: Wed, 15 Jul 2020 16:44:12 +0100
From: root <root@lordep>
Message-Id: <202007151544.06FFICD0003815@lordep>
Subject: envio e entrada de e-mail
To: <teste1@gri.pt>
X-Mailer: mail (GNU Mailutils 3.4)

Este é um e-mail de teste ao envio e entrada de correio eletrónico de root para teste1
? quit
Saved 1 message in /home/teste1/mbox
Held 0 messages in /var/mail/teste1
teste1@lordep:~$
```

## Blackhole list

A blackhole list é um recurso público fornecido para ajudar a reduzir o volume de publicidade não solicitada. Origens de email e hosts conhecidos estão listados num banco de dados consultável na Internet. São inseridos lá por pessoas que receberam publicidade indesejada de algum endereço de e-mail. Por vezes, os principais domínios estão na lista devido a falhas no controlo de spam. Embora algumas pessoas se queixem de escolhas específicas feitas pelos gestores da lista, esta continua bastante popular e as divergências geralmente são resolvidas rapidamente.

Esta funcionalidade do sendmail testará o endereço de origem de cada mensagem recebida em relação à blackhole list em tempo real para determinar se a mensagem deve ser ou não aceite.

Para configurar a funcionalidade Blackhole List em tempo real, adicionámos a seguinte declaração no ficheiro *sendmail.mc*:

FEATURE(rbl)



## Smart Host

Por vezes, um host encontra correio eletrónico que não é possível ser enviado diretamente para o host de destino desejado. É, por vezes, conveniente ter um host único numa rede, com o propósito de gerir transmissões de e-mail para hosts remotos de maior dificuldade em conectar, em vez de ter cada local host a tentar efetuar este processo individual e independentemente.

Hosts que desempenham este papel de host central são chamados de smart hosts. Ao possuir um destes smart hosts responsáveis pela aceitação de e-mail, é possível enviar qualquer tipo de e-mail e este encarregar-se-á do roteamento e transmissão desse e-mail para o destino desejado.

```
gedit /etc/mail/sendmail.mc  
define(`SMART_HOST', `smtp.gri.pt')
```

Configurações adicionais:

```
define(`confAUTH_OPTIONS', `A p')dnl  
TRUST_AUTH_MECH(`EXTERNAL DIGEST-MD5 CRAM-MD5  
LOGIN PLAIN')dnl  
define(`confAUTH_MECHANISMS', `EXTERNAL GSSAPI DIGEST-  
MD5 CRAM-MD5 LOGIN PLAIN')dnl
```

Como o servidor SMTP exige autenticação, adicionámos a seguinte linha após as anteriores:

```
FEATURE(`authinfo', `hash -o /etc/mail/authinfo/gri-smtp-auth.db')dnl
```





Instituto Superior de Contabilidade e Administração de Coimbra

```
Activities Text Editor
gui 01:12
*sendmail.mc
Save
dnl # The access db is the basis for most of sendmail's checking
FEATURE('access_db', , 'skip')dnl
dnl #
dnl # The greet_pause feature stops some automail bots - but check the
dnl # provided access db for details on excluding localhost...
FEATURE('greet_pause', '1000')dnl 1 seconds
dnl #
dnl # Delay_checks allows sender<->recipient checking
FEATURE('delay_checks', 'friend', 'n')dnl
dnl #
dnl # If we get too many bad recipients, slow things down...
define('confBAD_RCPT_THROTTLE', '3')dnl
dnl #
dnl # Stop connections that overflow our concurrent and time connection rates
FEATURE('conncontrol', 'nodelay', 'terminate')dnl
FEATURE('ratecontrol', 'nodelay', 'terminate')dnl
dnl #
dnl # If you're on a dialup link, you should enable this - so sendmail
dnl # will not bring up the link (it will queue mail for later)
dnl define('confCON_EXPENSIVE', 'True')dnl
dnl #
dnl # Dialup/LAN connection overrides
dnl #
include('/etc/mail/m4/dialup.m4')dnl
include('/etc/mail/m4/provider.m4')dnl
dnl #
dnl # Smart host
define('SMART_HOST', 'smtp.gri.pt')dnl
define('confAUTH_OPTIONS', 'A p')dnl
TRUST_AUTH_MECH('EXTERNAL DIGEST-MD5 CrAM-MD5 LOGIN PLAIN')dnl
define('confAUTH_MECHANISMS', 'EXTERNAL GSSAPI DIGEST-MD5 CrAM-MD5 LOGIN PLAIN')dnl
FEATURE('authinfo', 'hash -o /etc/mail/authinfo/gri-smtp-auth.db')dnl
dnl #
dnl # Default Mailer setup
MAILER_DEFINITIONS
MAILER('local')dnl
MAILER('smtp')dnl
Plain Text Tab Width: 8 Ln 109, Col 39 INS
```

## Make Auth Directory

Para guardar os detalhes da conta foi necessário criar uma nova diretoria.

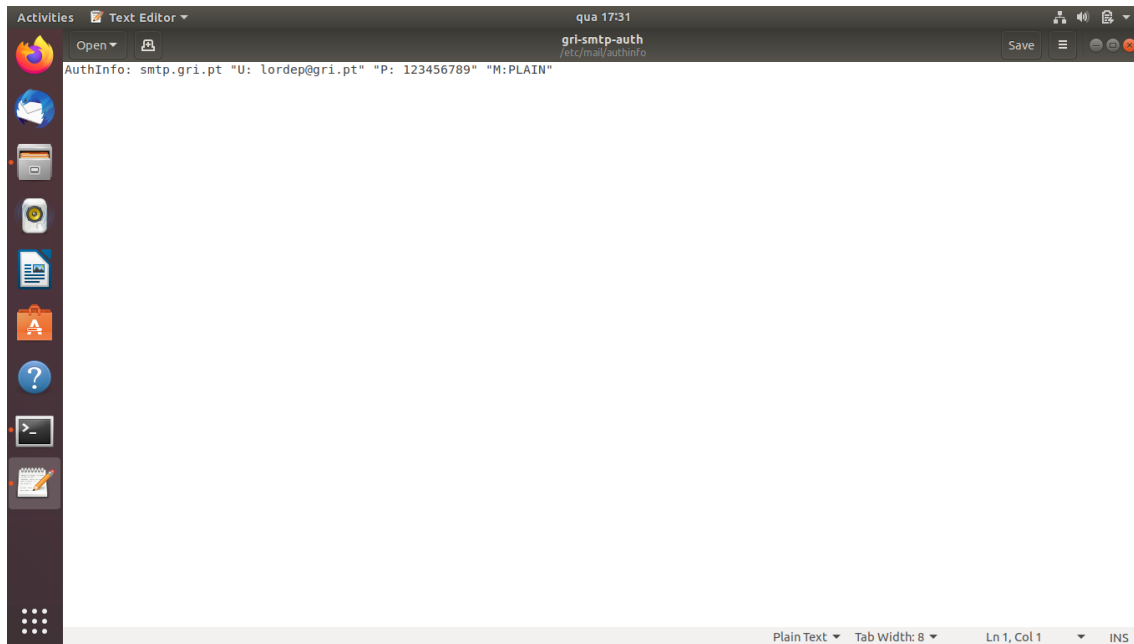
```
mkdir -m 700 /etc/mail/authinfo
```

Na diretoria acima criamos um ficheiro chamado **gri-smtp-auth** onde vamos inserir a conta criada.

```
cd /etc/mail/authinfo
touch gri-smtp-auth
gedit /etc/mail/authinfo/gri-smtp-auth
```

Especificaremos os dados de autenticação no servidor SMTP adicionando a seguinte linha:

```
AuthInfo: smtp.gri.pt "U: lordep@gri.pt " "P: 123456789" "M: PLAIN"
```



Onde smtp.gri.pt é o nome do nosso servidor smtp, lordep@gri.pt é a conta do usuário smtp que usaremos para enviar e-mails e password é a senha da conta smtp usada. M: PLAIN refere-se ao método usado para enviar os cabeçalhos de autenticação.

Foi necessário criar o mapa da base de dados a partir do arquivo gri-smtp-auth usando o comando makemap:

```
makemap hash /etc/mail/authinfo/gri-smtp-auth <
/etc/mail/authinfo/gri-smtp-auth
```

Isto irá criar o ficheiro gri-smtp-auth.db indicado anteriormente em FEATURE(`authinfo', `hash -o /etc/mail/authinfo/gri-smtp-auth.db')dn1



## Masquerading

É necessário estabelecer que gri.pt corresponde à máscara que usaremos em todos os emails que emitimos do nosso servidor. Para tal fizemos as seguintes alterações:

```
gedit /etc/mail/sendmail.mc
```

MASQUERADE\_DOMAIN(gri.pt)dnl → definir o domínio ao qual os hosts mascarados pertencem.

MASQUERADE\_AS(`gri.pt')dnl → domínio final

FEATURE(masquerade\_envelope)dnl → faz com que o sendmail também aplique o masquerading ao envelope do endereço do emissor.

FEATURE(masquerade\_entire\_domain)dnl → garantir que todos os hosts do domínio sejam mascarados.

dnl EXPOSED\_USER(`root')dnl → comentámos de forma à conta root ser também mascarada e não exposta.

```
m4 /etc/mail/sendmail.mc > /etc/mail/sendmail.cf
```

```
service sendmail restart
```

## Teste masquerading

```
root@lordep:/home# mail -s "masquerading teste" teste1@gri.pt
Cc:
Este é um e-mail de teste à masquerading de root para teste1
root@lordep:/home# su - teste1
teste1@lordep:~$ mail
"/var/mail/teste1": 1 message 1 new
>N 1 root qua Jul 15 17:49 15/570 masquerading teste
? 1
Return-Path: <root@gri.pt>
Received: from lordep (localhost [127.0.0.1])
  by lordep (8.15.2/8.15.2/Debian-10) with ESMTP id 06FGnQ5q004325
  for <teste1@gri.pt>; Wed, 15 Jul 2020 17:49:26 +0100
Received: (from root@localhost)
  by lordep (8.15.2/8.15.2/Submit) id 06FGnQ53004324;
  Wed, 15 Jul 2020 17:49:26 +0100
Date: Wed, 15 Jul 2020 17:49:26 +0100
From: root <root@gri.pt>
Message-Id: <202007151649.06FGnQ53004324@lordep>
Subject: masquerading teste
To: <teste1@gri.pt>
X-Mailer: mail (GNU Mailutils 3.4)

Este é um e-mail de teste à masquerading de root para teste1
? quit
Saved 1 message in /home/teste1/nbox
Held 0 messages in /var/mail/teste1
teste1@lordep:~$
```



## Aliases – envio de mail local

Aliases de e-mail permitem que o e-mail seja direcionado para mailboxes com nomes alternativos num host de destino.

É no ficheiro aliases que estes nomes se encontram registados. Sendmail consulta este ficheiro sempre que precisa de determinar como lidar com uma mensagem de e-mail recebida. Caso encontre uma entrada neste ficheiro que corresponda ao utilizador em questão na mensagem de e-mail, redireciona a mensagem para onde quer que a entrada descreva.

```
gedit /etc/mail/aliases
```

```
root: lordep
```

```
teste1: teste2, teste3
```

```
teste3: teste4
```

Para cada envelope que lista um usuário local como um destinatário, o sendmail procura esse mesmo nome no ficheiro de aliases. (Usuário local é um qualquer endereço que normalmente seria entregue na máquina local.) Quando o sendmail está a processar o envelope e corresponde ao usuário com um dos nomes à esquerda no ficheiro de aliases, este substitui o nome do destinatário pelo texto à direita do carácter ':'. Por exemplo, lordep torna-se o novo destinatário dos e-mail destinados ao usuário root.

Após a substituição do nome, o novo nome é procurado e todo o processo é repetido até que não hajam mais correspondências encontradas. O nome teste1 é alterado a primeira vez para teste2 e teste3. De seguida, o processo repete-se, desta vez para teste2 e teste3, sendo este último alterado para teste4. Uma vez que não há entrada para teste4 nem para teste2 no ficheiro aliases, o e-mail é então entregue nas mailboxes de teste2 e teste4.

Após a criação ou modificação, é necessário o uso do comando *newaliases*, responsável pela geração de uma nova versão do ficheiro aliases para aliases.db:

```
newaliases
```



```
m4 /etc/mail/sendmail.mc > /etc/mail/sendmail.cf  
service sendmail restart
```

## Teste aliases:

```
Activities Terminal  
qua 18:14  
teste4@lordep: ~  
File Edit View Search Terminal Help  
root@lordep:~# mail -s "aliases teste" teste1@gri.pt  
Cc:  
Este é um e-mail de teste às aliases de root para teste1  
root@lordep:~# su - teste1  
teste1@lordep:~$ mail  
No mail for teste1  
teste1@lordep:~$ su - teste2  
Password:  
teste2@lordep:~$ mail  
"/var/mail/teste2": 1 message 1 new  
>N 1 root qua jul 15 18:13 15/561 aliases teste  
? quit  
Held 1 message in /var/mail/teste2  
teste2@lordep:~$ su - teste3  
Password:  
teste3@lordep:~$ mail  
No mail for teste3  
teste3@lordep:~$ su - teste4  
Password:  
teste4@lordep:~$ mail  
"/var/mail/teste4": 1 message 1 new  
>N 1 root qua jul 15 18:12 15/561 aliases teste  
? 1  
Return-Path: <root@gri.pt>  
Received: from lordep (localhost [127.0.0.1])  
by lordep (8.15.2/8.15.2/Debian-10) with ESMTP id 06FHCxid004736  
for <teste1@gri.pt>; Wed, 15 Jul 2020 18:12:59 +0100  
Received: (from root@localhost)  
by lordep (8.15.2/8.15.2/Submit) id 06FHCxit004730;  
Wed, 15 Jul 2020 18:12:59 +0100  
Date: Wed, 15 Jul 2020 18:12:59 +0100  
From: root <root@gri.pt>  
Message-Id: <202007151712.06FHCxit004730@lordep>  
Subject: aliases teste  
To: <teste1@gri.pt>  
X-Mailer: mail (GNU Mailutils 3.4)  
...  
Este é um e-mail de teste às aliases de root para teste1
```

Por questões de segurança desativamos os comandos EXPN e VRFY:

Desativar comando VRFY: O PrivacyOptions=novrfy

Desativar comando EXPN: O PrivacyOptions=noexpn

Desativar Ambos: O PrivacyOptions=goaway

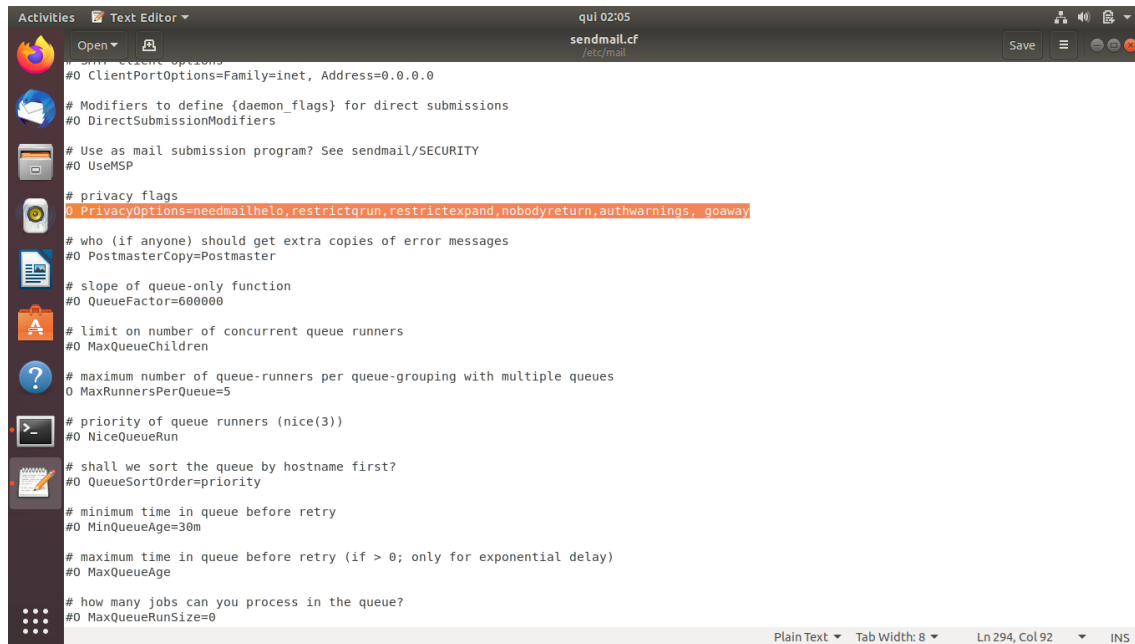
Para tal recorremos ao ficheiro sendmail.cf:

```
gedit /etc/mail/sendmail.cf
```

O PrivacyOptions=goaway



## Instituto Superior de Contabilidade e Administração de Coimbra



```
Activities Text Editor
Open
# ClientPortOptions=Family=inet, Address=0.0.0.0
# Modifiers to define {daemon_flags} for direct submissions
#0 DirectSubmissionModifiers
# Use as mail submission program? See sendmail/SECURITY
#0 UseMSP
# privacy flags
0 PrivacyOptions=needmailhelo,restrictqrun,restrictexpand,nobodyreturn,authwarnings,goaway
# who (if anyone) should get extra copies of error messages
#0 PostmasterCopy=Postmaster
# slope of queue-only function
#0 QueueFactor=600000
# limit on number of concurrent queue runners
#0 MaxQueueChildren
# maximum number of queue-runners per queue-grouping with multiple queues
0 MaxRunnersPerQueue=5
# priority of queue runners (nice(3))
#0 NiceQueueRun
# shall we sort the queue by hostname first?
#0 QueueSortOrder=priority
# minimum time in queue before retry
#0 MinQueueAge=30m
# maximum time in queue before retry (if > 0; only for exponential delay)
#0 MaxQueueAge
# how many jobs can you process in the queue?
#0 MaxQueueRunSize=0
Plain Text Tab Width: 8 Ln 294, Col 92 INS
```

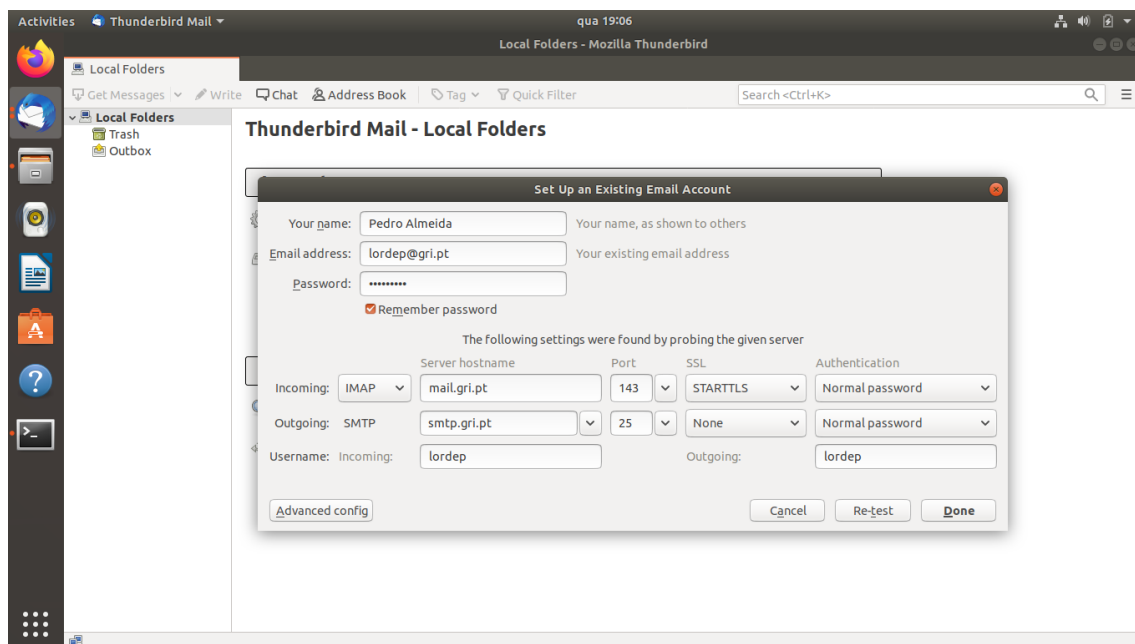


## Configuração do cliente de correio eletrónico

Depois de instalado e configurado um servidor de e-mail utilizado o Sendmail e o Dovecot, sem um cliente de e-mail pouco nos vale ter apenas o servidor. Configurámos, então, o Thunderbird para lidar com o envio e receção de e-mails.

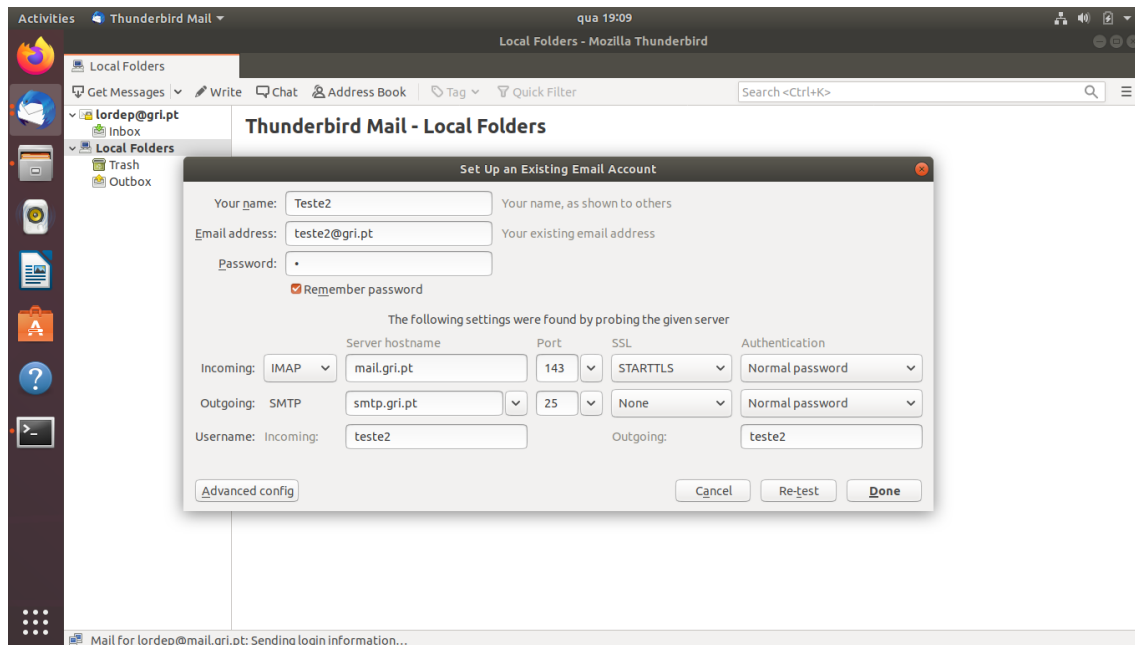
O Thunderbird, desenvolvido pela Mozilla Foundation, além de permitir enviar, receber e gerir todo o correio eletrónico, possui um filtro anti-spam embutido que previne eventuais ataques por via de e-mail.

Primeiramente adicionámos ao cliente o utilizador com o qual efetuámos login no Ubuntu, inserido, através da configuração manual os nomes dos servidores IMAP e SMTP, assim como os portos a usar, portos estes configurados em passos anteriores:

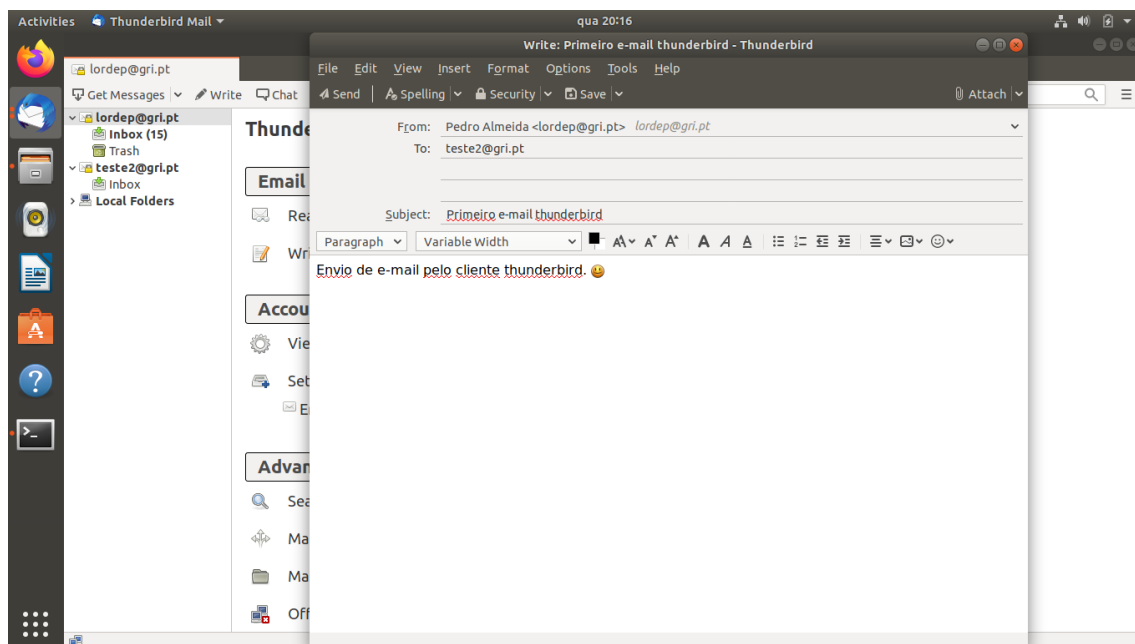




Para testar a comunicação entre duas contas de e-mail, adicionámos um novo e segundo utilizador, neste caso de nome Teste2, repetindo o processo do passo anterior:



Enviámos um e-mail de uma conta para a outra:

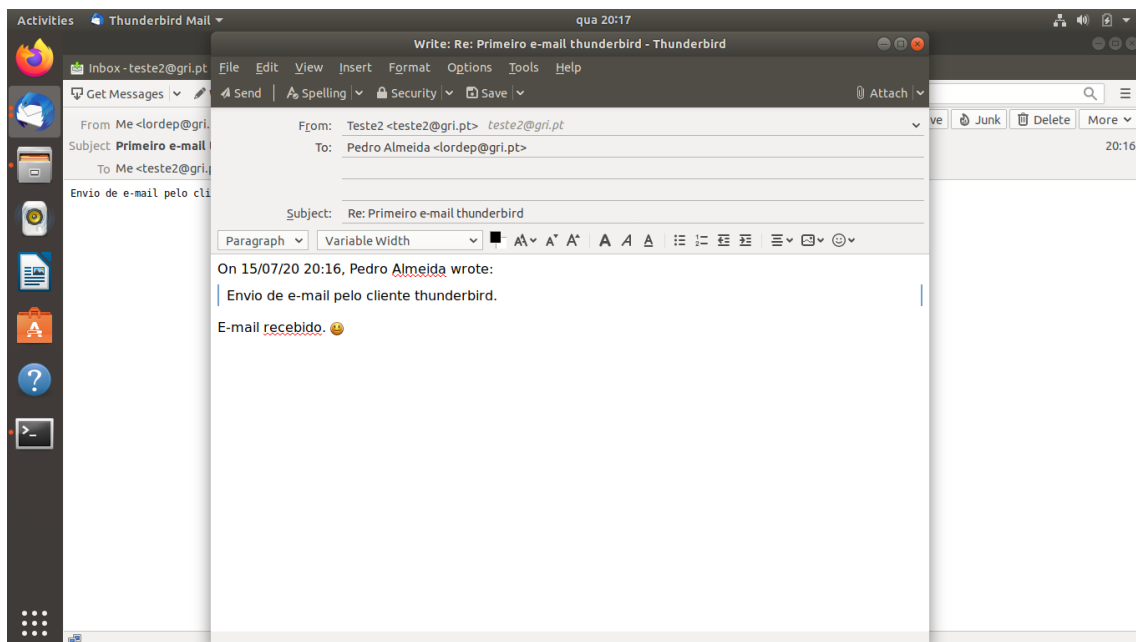
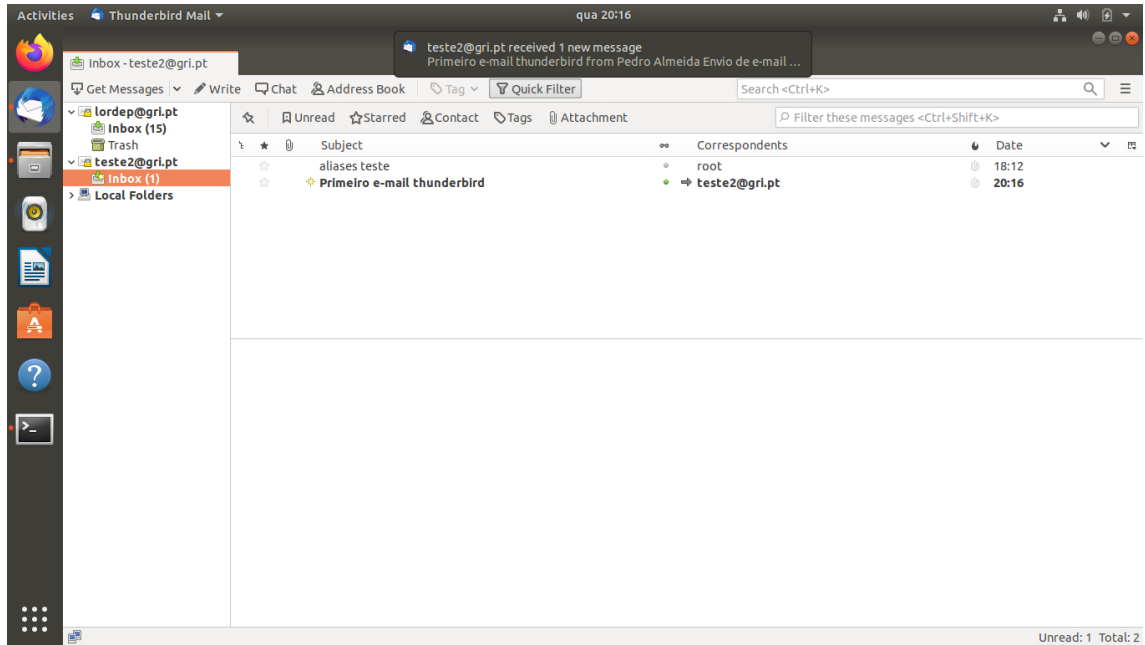






Instituto Superior de Contabilidade e Administração de Coimbra

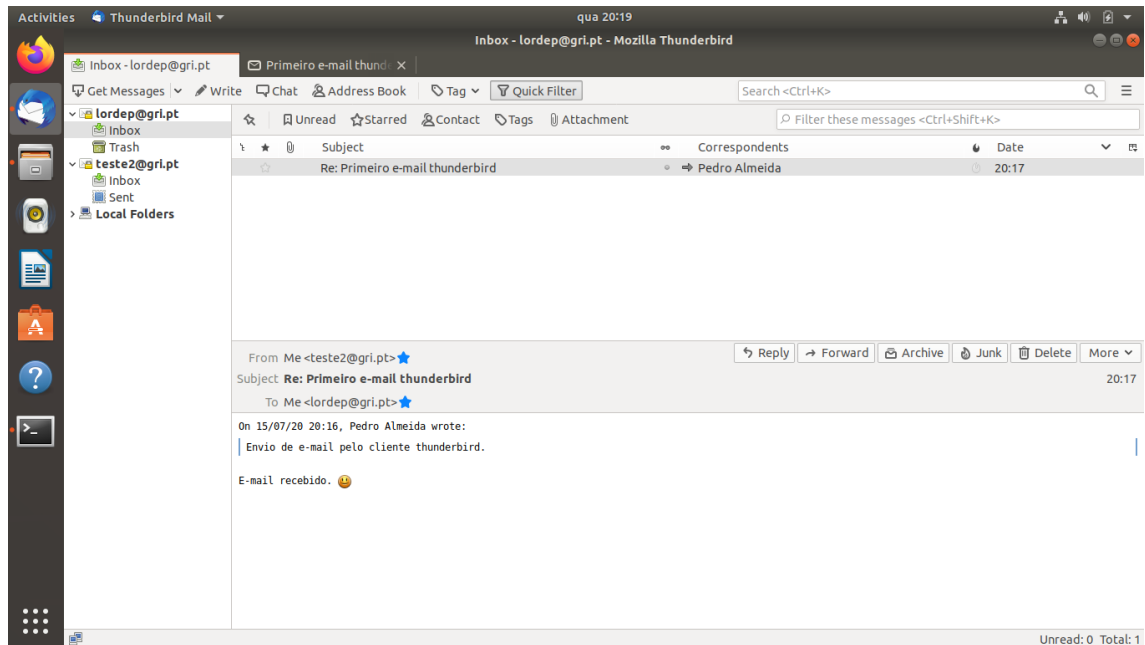
Verificámos que o e-mail foi recebido pelo utilizador Teste2 e respondemos a esse mesmo e-mail:





Instituto Superior de Contabilidade e Administração de Coimbra

Verificámos que o utilizador lordep também recebeu o e-mail de resposta por parte do utilizador Teste2:



E desta forma conclui-se que o servidor responde de forma rápida e eficaz, sem quaisquer erros ao envio e receção de e-mails.



## Conclusão

Este projeto foi dividido em 2 fases, sendo a primeira uma recapitulação da matéria lecionada na unidade curricular Gestão de redes informáticas, onde é explicado o funcionamento de um correio eletrónico, envolvendo todos os conceitos, protocolos e serviços adjacentes. Esta fase do projeto revelou-se bastante importante pois proporcionou-nos um maior entendimento dos conceitos necessários para a realização do trabalho final.

Na segunda parte foi explicado passo a passo o desenvolvimento da configuração de um servidor de correio eletrónico. Foi nesta fase que aplicámos os conhecimentos apresentados na fase inicial do projeto. Apesar da compreensão dos conceitos, deparámo-nos com várias dificuldades na implementação prática dos mesmos, o que dificultou e atrasou bastante o desenvolvimento do projeto.

Concluimos então que a configuração de um serviço de correio eletrónico é mais complexo do que esperávamos. Desta forma, a realização deste projeto desenvolveu significativamente as nossas capacidades e proporcionou-nos uma enorme bagagem de conhecimento do assunto em questão.