

# Correo Electrónico

Revisión: 2025-05-21

## SMTP

- E55.** Si usted desea entregar de forma directa un correo electrónico a una casilla del dominio `itba.edu.ar` ¿A qué servidores SMTP deberá conectarse?
- E56.** Intente conectarse a otros servidores SMTP que no sean administrado por el ITBA o a la red en la que se encuentra (por ejemplo `smtp.fibertel.com.ar`).
- ¿Puede?
  - Si no puede establecer la conexión: ¿Por qué cree que el administrador del ITBA configuró la red de esa manera?
  - Si le es posible intente enviar un correo electrónico a su cuenta del ITBA. ¿Es posible?
  - ¿Qué significa la configuración de relay?
  - Teniendo en cuenta todo el análisis anterior. ¿Con qué políticas configuraría su servidor SMTP?
- E57.** Instale **dovecot** y **postfix**. Configurarlo para que los correos enviados a su casilla puedan ser obtenidos mediante el protocolo POP3.

Envíe un correo a su cuenta de correo del ITBA con título "PDC Ejercicio 1", y cuerpo "hola mundo".

- Use en el `MAIL FROM:` su cuenta de correo del ITBA. Valide que el correo llegue. Revise desde el MUA el "fuente de correo". ¿El correo fue modificado?
- Haga los cambios necesarios para asegurar que en el campo `To:` que muestra el MUA se vea la dirección del destinatario.
- Manteniendo su dirección de correo del ITBA en el `MAIL FROM:` agregue en el mensaje una cabecera `From` con la dirección `alguien@example.org`.
  - ¿Con qué dirección se muestra en el MUA el remitente?
  - ¿Qué ve en las cabeceras del mensaje fuente?
- Manteniendo los cambios realizados. ¿Que sucede si en el mensaje (`DATA`) se escribe una cabecera `To: pepe@example.org`. ¿El mensaje le llega a usted?
- Manteniendo los cambios realizados anteriormente, en `RCPT TO:` use una dirección inexistente.
  - ¿Se le notifica que no se pudo entregar el correo?
  - ¿Qué dirección se utiliza para notificar los rebotes?



## Sugerencia

Es posible que en ciertos casos los correos no lleguen a destino, o lleguen con retraso. Verifique los logs, y comprenda las razones. Puede utilizar **mailq** y **postsuper** para verificar el estado de la cola de salida.

- E58.** Repita el ejercicio anterior, pero utilizando como destino una dirección servida por su SMTP local.

- Revise los correos electrónicos desde el filesystem (*maildir* o *mailbox*).
- Revise los correos electrónicos utilizando el protocolo POP3 utilizando netcat.

- E59.** {W} Utilizando un MUA envíe un correo electrónico que contenga de mensaje el texto

```
hólá mundó ñ
```

y contenga como adjunto la imagen ubicada en <https://www.itba.edu.ar/wp-content/uploads/2020/07/Marca-ITBA.jpg>.

- E60.** Verifique como realiza la transacción SMTP el MUA. Detecte diferencias con las que usted ha realizado en los ejercicios anteriores.

Utilizando el MUA también obtenga utilizando POP3 y verifique la transacción.

- E61.** Analizar el funcionamiento de las siguientes técnicas anti-spam. ¿Cuales son sus ventajas y desventajas?

- SPF
- Domain keys
- Grey Listing
- Filtros Bayesianos

- E62.** Según la configuración SPF de `itba.edu.ar`: ¿desde qué direcciones IP se pueden enviar correos a servidores que implementen SPF?

## Codificaciones BASE

- E63.** Codifique en `base64` de forma manual (sin usar ningún programa de computadora) el siguiente mensaje: HO LA (el mensaje está compuesto por 5 caracteres: 'H', 'O', ' ', 'L', 'A').

- E64.** Decodifique de forma manual (sin usar ningún programa de computadora) el siguiente mensaje que fue codificado con `base64` Y2hhdQ==

- E65.** Decodifique con el comando **base64** el siguiente mensaje

```
/9j/4AAQSkZJRgABAgAAZABkAAD/7AARRHVja3kAAQAEAAAALwAA/  
+4ADkFkb2J1AGTAAAAAf/bAIQACQYGBgcGCQcHCQ0JBwkNDwsJCQsPEg4ODw4OEhEODw4  
ODw4RERQVFxUUERSbHR0bGycmJiYnLCwsLCwsLCwsLAEKCQkKCwoMCgoMDw0ODQ8TDg4OD
```

hMVDg4QDg4VGxQRERERFBsYGhcXFxoYHh4bGx4eJiYkJiYsLCwsLCwsLCws/8AAEQtALwB  
4AwEiAAIRAQMRaf/EAJQAAAIDAAMBAAAAAAAAAAAAAAGBAUHAQITIAwEAAgMBAAAAAAAA  
AAAAAAAAgMAAQFEAACAQMDAgUBBQYHAQAAAABAgMRBAUAEGYhEzFBiHQM1FhI7M2cYG  
RQjNTocFysnN0FSQRAAEDAwIEBAUFAAAAAAAAERAqMAIRIxQVFhEwSBoSIUkbEyYgXh8  
VIjFF/aAAwDAQACEQMRAD8A3HRrrJIkaNJIwSNAwd2NAAOpJ1nWW5PyLkObu+NYi3nsbH  
YI1y0Q09ZG9cdxVT/AEHClar9ta+WmxQukJQgBoVzjoBQveGpuTYAVorOifUwWvhU01yCC  
KjqNefuzW9tJYvjbKSACYrKezElzu7jNcwRiRzv60aaByPu1Kg5Bf4vk2WzMN1LFisXts4  
7ZXPbuJYk9vb2+3wp+GXY+SjWv/OJYHNkuQShanDEa7rSPCoULffFj8K3jRpX4Fy+XkmII  
nvYBaX53eitFmRKAzwKx3bKmh+/z1nPzDmcvZ8uWGzvri3h9pE3bildFqWkqaKRpMPaPkn  
MBIA4AruLUx8zWxiQBQa27RrFr+35vleF8Ynwr31xL2rg3csMrBie7RO428E+HTSXksnzL  
F3j2WQvr63uowC8TzvUBhuHg58Rp8f47qKBMzIFwLd/SU0pb+5x1YUIF9rha9O6NYJxzFf  
Jk2Rx144yL455oJWkadijQ11YsQZPDbqL8h570W3M8rDb5C5ihSUBI45nVVGxD0UNTVD8d  
1J02zMd6S4kXRCiVD3KNyLClpevQujXn3keP5zxe2tr65z7SCdtsa295Kzg7d1SrU6a0z4  
o5NluQcf1kyh7k9rMYFuSKd1dq6qADctaHS5uyMcXWbI17VS1qNk4c/AtLSi3p20axD5f  
5jeTchXE466khgxy7zmhdk3TvQsCVIrsFB+2uuPiLmN5ByM4vJXUk1vkV2RGZ2fZ01s1N5  
NNwqv7aaL/Pk9v11H05YJdP2ofct6nTTdF51uGjRo1hrRSX8q5eysuNGyu55rZcm/t+9bo  
sjKoG99ysyVU02mhr11nfHchPj8XdY+wuD1mulNta3Fo6x3V1DJ/U7UM6rNVmpVVO3p416  
60v5BTBm0s3zFmt5GJssSmNpCpYdWVU1iPgPv0p2Fvw97hbjG4WB5bdoJYpAjL/UI7cibr  
zqA1NdPtnNhB4ljijsYYqr2NZZQTKuQFk5pTJhvjWxSAT5uV7zJzNbzXTq5WMzWtRFIB  
9W7afVU9dKPN+FLgTbTWsT5DHb5XhtZiFjf1IS8k97KSgZdoUKOlaUJ+1xg5tdykBoliLq  
hh3Rr+IZBbMqLS68St0h/j9mo82TtuUi3xWUs4pIZZAYRzR1A1VZWTpHd7vpRvUOg0Mcnc  
MkzkKt1cBwTarc2MtxaL7frWaYLLR4z9rncrmRPeo6o1vZJ3gY2/D7TSfhwogB8ErTy1L  
+bP1mv/AE4f90mr9MfxATGN8DBFIoDKSjEVMrW6DcLzaKyIRUnoOumpWu+NZnJn/wBHHQ3  
WURRGN8e0FEAf0y+9ERUbm618RT7Naus0Ts1DHelhfp02RNhSSwmMsLhcgjWmb4v/QWI/  
wCN/wA2TWQfLv67yH+iD8pdafj+UWGMw9omNWOPHATbIUhIaKKKVY3Yq91uJLSAhRVtU19  
NxXN35ub3G29zkJ6bndQtVWMMOpd/eqi/h+vfu0jt3GPuZZ3MJa7Kw1CuW9M1AfG2MEKE+V  
PfEf0tiP+nB+WusD+S/1zmKf31/LTw2PJ+zjCLJI1t7T20NvD2wN0c7m3h9RuyE6oahyC  
NUFzHxXMxZLPXOKhnaHtvNK8MgeVpCsY2gXVO1Rqu0cYZpJHNJDvTbVXEEVJgHsawEAi9+  
VIPJo/jpcZG3HZrx8nuXes4Pb2U9dd6j91NM3xjyq9wvE87dXRL4ywCNZK3h7mXcO0h89z  
bSR5amdgjgix2s0Wht5GuXKJH2XJuH29a+7b+5/gdWE2a41kcaMXJYQnH2rGV7UW/ZSPtt  
2zIFF0m8+sEAVJBrrRJTHxCIske0uBLnoXAA7fKltaj8w5oKIA3Ssrw1vn8nlzkLGyfJ3k  
Mou51CGRS5ffFWUCnQtrjNW2exuWF9f2T4y8n1N3AmwxqGD76xA16Bta/g8tgMFixe46CDH  
Rxt4tnLGIGL71Fe5JW6b0qrV/fqPkOT4fPyW8ORx0V6yuFt+9bbgvdBLGvuOn0ddH7x/UJ  
EP9YGP3J8tarotxHr9Rvyp14rnYc/gLPKx0BnjHdQfyrr6ZF/cw0ag8F1xcmNuFx1rHZwx  
3DI8EUYiG/ah3lRJL9SkeejXIwb7jHE45fTvjq1bMj01UKmvOry7soroJ3GZTGdy1SAa/b  
1B1FTA2cZLI0iFtu4qVBO36a0Xy8tWWjSg5wCAo0aDhVVNhrRRET3pNsifAu07G61JBUDNo  
8x5aqFuYVeCdMfk060/su4qAPErFdzsdoIj3P1I6eemzRomyEahfFKos4WpTuvbKZGfHXs  
rR2zzhtituUsUaAer9TEk7fs66+dxJbqtxC2LzYYbVbhaKjJIT229ulU+paj+h3acNGiE  
32+ZqsOf1Sm3tFnt2GPvC800R7mxQYeREFuk7PFe5Qn7jroHsiFjXG32x7toCrRBQaEFbk  
Ap1Vq1BOm/Rqdb7T8amHPypZyEEVrLd26WF3cWCH3UzRbCk0jMQI9pX1udtdfL/5zCr/  
+dek3XYSaPatazBqCX0+EZWjE/Tpr0aoS208zUw5+VJ8jWB3gYy9e1yLOjRAb40HomWsfW  
PxC/wCWpt1b2V3LJGbW8i9tc9mJpYwqnaDtuiyV+joQD4/x0x6NQy2sCPGoGc/KqxMBZIn  
bRnWP1ehSoX1/X02/zeeuH47j3VVfcyp9AO0hf2DZ01aaNB1H/wAjRYt4VGsrCCyjaOCu1  
2LkGn1HxPpA8dGpOjVZFclvxqIESv/Z

- a. ¿Qué representa el mensaje original?
- b. ¿Cuantos bytes ocupa el mensaje original?

- c. ¿Cuantos bytes ocupa el mensaje codificado?
  - d. Dado un mensaje cualquiera: ¿Puede predecir cuanto ocupará codificado en base64?
  - e. Teniendo en cuenta todo esto, si su proveedor de correo electrónico le comunica que podrá recibir correos electrónicos de hasta 10MB y alguien le quiere enviar un correo con una imagen cuyo tamaño es de 9MB. ¿Llegará?
- E66.** El ejercicio anterior muestra que es posible transportar en una hoja impresa `contenido binario'. ¿Elegiría la codificación *base64* para ése medio de transporte? Compare los criterios de selección del alfabeto de la Sección 3.4 del [RFC4648].
- E67.** Se encuentra diseñando una aplicación que utiliza HTTP. Desea exponer un recurso que contiene un identificador de un ancho fijo de 160 bits. La *URL Template* [RFC6570] podría ser `http://example.org/docid/{id}`. La intención del ejercicio es discutir diferentes formas de codificación del parámetro `{id}`.

Los 160 bits están contenidos en 20 bytes. Un ejemplo de este identificador representado en bytes sin signo podría ser:

```
0x42 0xa4 0x4e 0xe4 0x3f 0x1d 0x63 0xa9 0xf6 0x57
0x13 0xcb 0xa7 0x6d 0x42 0x30 0x3c 0xcc 0x35 0xa7
```

Que podría transportarse en una URL sin problema utilizando el *Percentage Encoding* (sección 2.1 de [RC3986]) de la siguiente forma:

```
%42%a4%4e%e4%3f%1d%63%a9%f6%57%13%cb%a7%6d%42%30%3c%cc
%35%a7 (60 bytes/caracteres ASCII)
```

Dichos bytes se podrían interpretar como un número entero con signo. Si se interpretan los bytes con una codificación *big endian*, complemento a 2, y se representa dicho número en base 10 se podría representar como:

```
380457585513508727835503478909238945443373397415 (48 bytes/caracteres ASCII)
```

Una representación hexadecimal más compacta podría ser:

```
42a44ee43f1d63a9f65713cba76d42303ccc35a7 (40 bytes/caracteres ASCII)
```

Decide aprovechar la codificación *base64* logrando una codificación de ancho fijo de 28 bytes:

```
QqRO5D8dY6n2VxPLp21CMDzMNac=
```

Quedando satisfecho con esta longitud decide implementarlo (un ejemplo de URL podría ser `http://example.org/docid/QqRO5D8dY6n2VxPLp21CMDzMNac=`).

- a. ¿El uso de base64 en URLs es una buena decisión o debe tomar algún recaudo?
- b. Utilizando esta codificación, ¿puede ahorrarse algún byte más al sintetizar las URLs?

## MIME

- E68.** ¿Por qué necesitamos de métodos de codificación como *Quoted Printable* o *Base64* en el envío de correos?
- E69.** Decodifique el mensaje codificado en el siguiente correo

```
MIME-Version: 1.0
Content-Type: multipart/alternative;
boundary="Boundary-01=_bHUWOieIFHJ6aoah"
Content-Transfer-Encoding: 7bit
--Boundary-01=_bHUWOieIFHJ6aoah
Content-Type: text/plain;
charset="iso-8859-1"
Content-Transfer-Encoding: quoted-printable

h=F3l=E1 c=F3m=F3
t=E9 v=E1
=2D-=20
Buenos Aires, Argentina
--Boundary-01=_bHUWOieIFHJ6aoah
```

- E70.** Utilizando un MUA, cree un correo electrónico cuyo mensaje principal esté formateado en HTML (agregue links; e imágenes externas). Adjunte también un archivo TXT, una imagen JPG, y una imagen PNG. Guárdelo como Draft, y vea su “código fuente”.
- a. Revise su estructura MIME
  - b. ¿Qué es cada parte MIME? Si el mensaje presenta una versión txt/plain del texto que usted escribió analice porque lo incluye si usted escribió un HTML.
  - c. Fowardee el correo draft como un adjunto MIME. ¿Cuales partes MIME que contienen otras partes?
  - d. ¿Para que sirve el header Content-ID?
  - e. ¿Para que sirve el header Content-Disposition?
  - f. Investigue el formato de los siguientes Media-Type:
    - i. multipart/digest
    - ii. multipart/mixed
    - iii. message/rfc822
    - iv. multipart/alternative
    - v. multipart/related

vi. multipart/report

# Bibliografía

[dovecot(1)] *dovecot - a secure and highly configurable IMAP and POP3 server.*

Dovecot is an open source IMAP and POP3 server for Linux/UNIX-like systems, written with security primarily in mind. Dovecot is an excellent choice for both small and large installations. It's fast, simple to set up, requires no special administration and it uses very little memory.

[postfix(1)] *postfix - Postfix control program.*

The postfix(1) command controls the operation of the Postfix mail system: start or stop the master(8) daemon, do a health check, and other maintenance.

[base64(1)] *base64 encode/decode data and print to standard output.*

Base64 encode or decode FILE, or standard input, to standard output.

[RFC1939] *Post Office Protocol - Version 3* [<https://tools.ietf.org/html/rfc1939>] . The Internet Engineering Task Force. May 1996.

The Post Office Protocol - Version 3 (POP3) is intended to permit a workstation to dynamically access a maildrop on a server host in a useful fashion.

[RFC2045] *Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies* [<https://tools.ietf.org/html/rfc2045>] . The Internet Engineering Task Force. November 1996.

This initial document specifies the various headers used to describe the structure of MIME messages

[RFC3986] *Uniform Resource Identifier (URI): Generic Syntax* [<https://tools.ietf.org/html/rfc3986>] . The Internet Engineering Task Force. January 2005.

A Uniform Resource Identifier (URI) is a compact sequence of characters that identifies an abstract or physical resource. This specification defines the generic URI syntax and a process for resolving URI references that might be in relative form, along with guidelines and security considerations for the use of URIs on the Internet. The URI syntax defines a grammar that is a superset of all valid URIs, allowing an implementation to parse the common components of a URI reference without knowing the scheme-specific requirements of every possible identifier. This specification does not define a generative grammar for URIs; that task is performed by the individual specifications of each URI scheme.

[RFC4648] *The Base16, Base32, and Base64 Data Encodings* [<https://tools.ietf.org/html/rfc4648>] . The Internet Engineering Task Force. October 2006.

This document describes the commonly used base 64, base 32, and base 16 encoding schemes. It also discusses the use of line-feeds in encoded data, use of padding in encoded data, use of non-alphabet characters in encoded data, use of different encoding alphabets, and canonical encodings.

[RFC5321] *Simple Mail Transfer Protocol* [<https://tools.ietf.org/html/rfc5321>] . The Internet Engineering Task Force. October 2008.

This document is a specification of the basic protocol for Internet electronic mail transport. It consolidates, updates, and clarifies several previous documents, making all or parts of most of them obsolete. It covers the SMTP extension mechanisms and best practices for the contemporary Internet, but does not provide details about particular extensions. Although SMTP was designed as a mail transport and delivery protocol, this specification also contains information that is important to its use as a "mail submission" protocol for "split-UA" (User Agent) mail reading systems and mobile environments.

[RFC5322] *Internet Message Format* [<https://tools.ietf.org/html/rfc5322>] . The Internet Engineering Task Force. October 2008.

This document specifies the Internet Message Format (IMF), a syntax for text messages that are sent between computer users, within the framework of "electronic mail" messages. This specification is a revision of Request For Comments (RFC) 2822, which itself superseded Request For Comments (RFC) 822, "Standard for the Format of ARPA Internet Text Messages", updating it to reflect current practice and incorporating incremental changes that were specified in other RFCs.

[RFC6570] *URI Template* [<https://tools.ietf.org/html/rfc6570>] . The Internet Engineering Task Force. March 2012.

A URI Template is a compact sequence of characters for describing a range of Uniform Resource Identifiers through variable expansion. This specification defines the URI Template syntax and the process for expanding a URI Template into a URI reference, along with guidelines for the use of URI Templates on the Internet.

[RFC7208] *Sender Policy Framework (SPF) for Authorizing Use of Domains in Email, Version 1*, [<https://tools.ietf.org/html/rfc7208>] . The Internet Engineering Task Force. April 2014.

Email on the Internet can be forged in a number of ways. In particular, existing protocols place no restriction on what a sending host can use as the "MAIL FROM" of a message or the domain given on the SMTP HELO/EHLO commands. This document describes version 1 of the Sender Policy Framework (SPF) protocol, whereby Administrative Management Domains (ADMDs) can explicitly authorize the hosts that are allowed to use their domain names, and a receiving host can check such authorization. This document obsoletes RFC 4408.