

# M1.3-Correo-electronico.pdf



alfonpx



Redes de Comunicaciones II



3º Grado en Ingeniería Informática



Escuela Politécnica Superior  
Universidad Autónoma de Madrid

**WUOLAH + BBVA**

Hazte **cliente de BBVA y...**  
**ahórrate 6 meses**  
**de suscripción**

**BOOM**

**1/6**

Este número es indicativo del riesgo del producto, siendo 1/6 indicativo de menor riesgo y 6/6 de mayor riesgo.

BBVA está adherido al Fondo de Garantía de Depósitos de Entidades de Crédito de España. La cantidad máxima garantizada es de 100.000 euros por la totalidad de los depósitos constituidos en BBVA por persona.

Ahora, si te abres una Cuenta Online en BBVA, te reembolsamos una de estas suscripciones durante 6 meses (hasta 9,99€/mes) al pagarla con tu tarjeta Aqua Débito

**NETFLIX**

**Spotify**

**HBOmax**

**Disney+**

**PlayStation Plus**

**DAZN**

Promoción solo para nuevos clientes de BBVA. Válida hasta el 30/06/2023. Estas empresas no colaboran en la promoción.

**Abre tu cuenta**



# Hazte cliente de BBVA y ... ahórrate 6 meses de suscripción

WUOLAH  
+ BBVA

NETFLIX

Spotify

HBOmax

Disney+

PlayStation.Plus

DAZN

Ahora, si te abres una Cuenta Online en BBVA, te reembolsamos una de estas suscripciones durante 6 meses (hasta 9,99€/mes) al pagarla con tu tarjeta Aqua Débito

Promoción solo para nuevos clientes de BBVA. Válida hasta el 30/06/2023. Estas empresas no colaboran en la promoción.



1/6

Este número es indicativo del riesgo del producto, siendo 1/6 indicativo de menor riesgo y 6/6 de mayor riesgo.

BBVA está adherido al Fondo de Garantía de Depósitos de Entidades de Crédito de España. La cantidad máxima garantizada es de 100.000 euros por la totalidad de los depósitos constituidos en BBVA por persona.

## M1.3 - Correo electrónico

Última actualización: 21/02/2023

1. Estructura
2. SMTP: puerto 25
  - 2.1. Un ejemplo
  - 2.2. Funcionamiento del protocolo
  - 2.3. Comparación con HTTP
3. Protocolos de acceso al correo
  - 3.1. POP: puerto 110
  - 3.2. IMAP: puerto 143
  - 3.3. HTTP: puerto de indias

Puede parecer que Gmail o Outlook son páginas web como las que hemos estudiado hasta ahora, pero esto no es cierto del todo.

Utilizar Internet para enviar mensajitos se lleva haciendo desde sus orígenes, por lo que este nuevo sistema postal se ideó muchísimo antes de poder imaginar que algún día utilizaríamos ordenadores para ver vídeos de gatos. Sin embargo, asentó las bases de cómo nos comunicamos hoy en día desde nuestros dispositivos.



El primer correo electrónico se envió en 1971.

Decía  
"QWERTYUIOP"

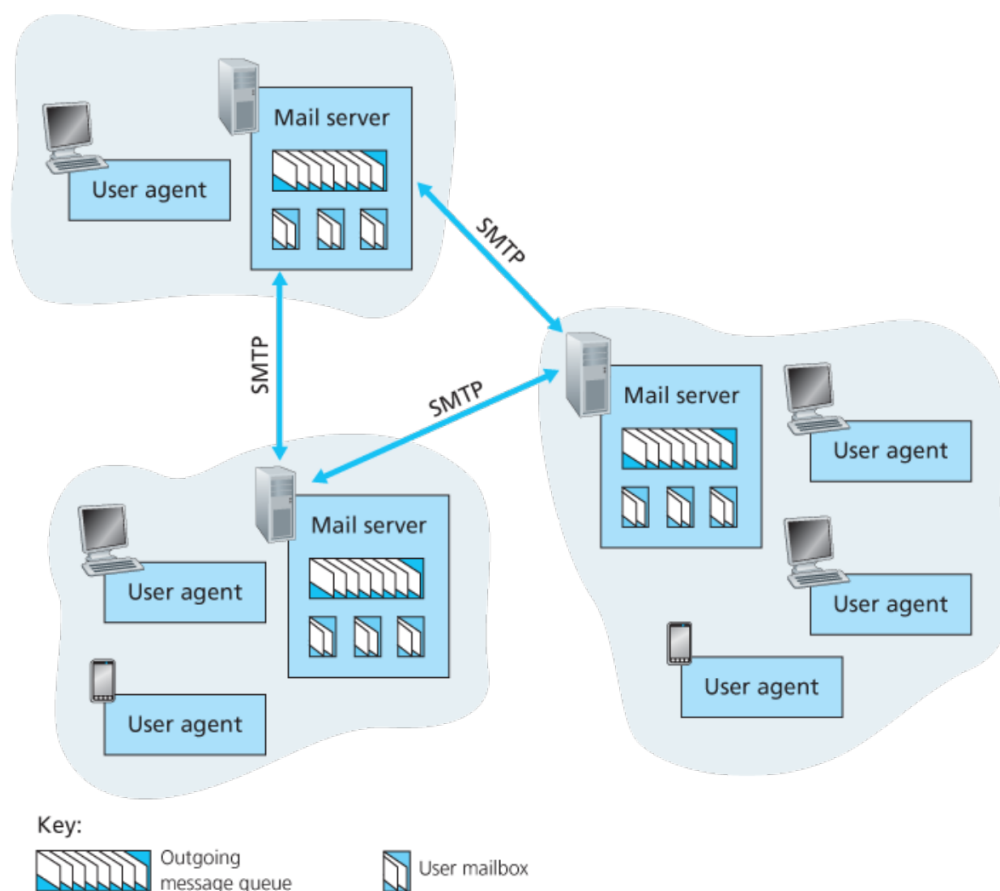
### 1. Estructura



Abre tu cuenta



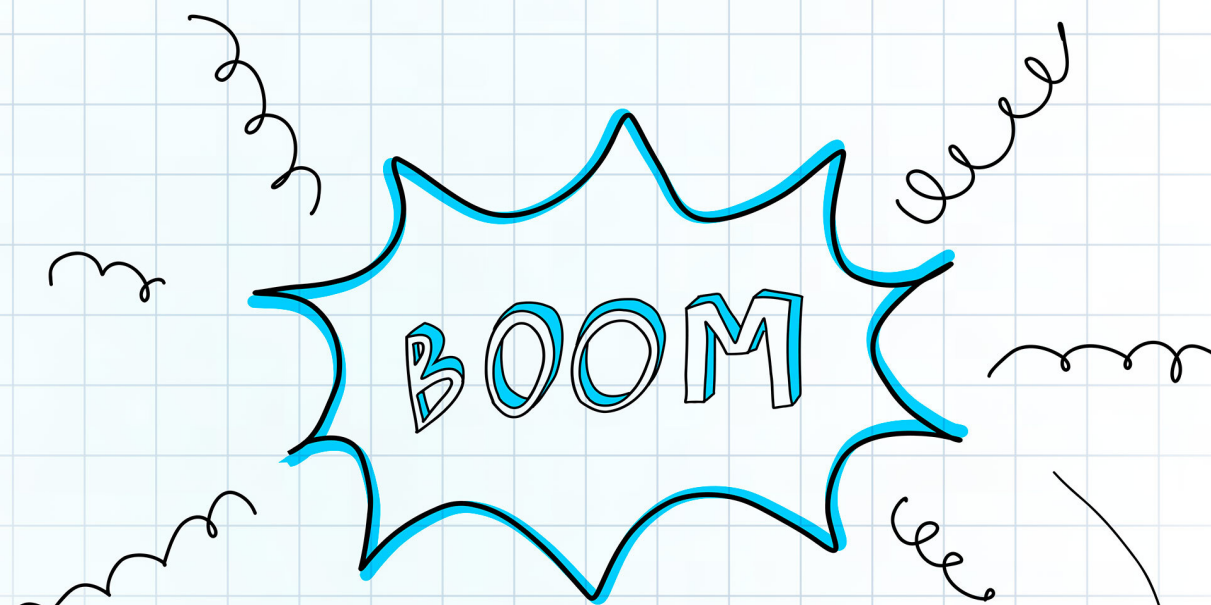
WUOLAH  
+ BBVA



Existen tres componentes principales para el sistema de correo electrónico:

- **Usuario (User Agent):** la aplicación desde la que consultas el correo electrónico, como Google Chrome (luego te conectas a Gmail, Outlook, Webmail, ...) o la app de Correo del iPhone. Se conecta al servidor de correo para enviar y recibir mensajes.
- **Servidor de correo (Mail server):** servidor que almacena todos los correos electrónicos y los organiza para todos los usuarios registrados. Por ejemplo, mis correos personales están en los servidores de Gmail, los de las cosas de la uni están en los servidores de la UAM (@estudiante.uam.es), etc.  
Tienen la característica de utilizar colas de mensajes, tanto para recibir correos como para enviarlos.
- **SMTP:** protocolo con el que se comunican los distintos servidores de correo electrónico.

Es importante observar que, a diferencia de lo que ocurría con HTTP, **siempre es necesario utilizar dos servidores:** uno es el remitente (en el que inicias sesión y



# Hazte cliente de BBVA y ... **ahórrate 6 meses** **de suscripción**

Ahora, si te abres una Cuenta Online en BBVA, te reembolsamos una de estas suscripciones durante 6 meses (hasta 9,99€/mes) al pagarla con tu tarjeta Aqua Débito

**NETFLIX****HBOmax**[Abre tu cuenta](#)

redactas el correo) y otro es el destinatario (en el que la otra persona inicia sesión y lee tu correo). En HTTP, el cliente (Google Chrome) mandaba la petición directamente al servidor web que le interesaba.

Existe el caso excepcional de que el remitente y el destinatario compartan servidor de correo; por ejemplo, si los dos utilizan Gmail. El diagrama se simplificaría, pero la idea es la misma. Asumiré de ahora en adelante que **NO** comparten servidor de correo.

## 2. SMTP: puerto 25

SMTP es el protocolo que se utiliza para el correo electrónico. ¿Otro protocolo? ¿Para qué tantos? ¿No se supone que HTTP ya te permitía enviar todo lo que quieras? Paciencia.

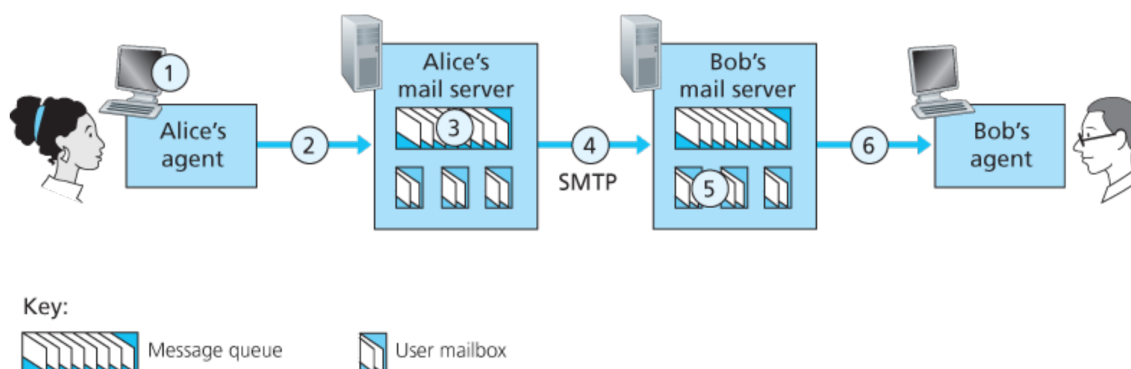
*Calma... Calma, calma, calma*

*~ Magüi*

Por ahora, lo único que te puedo decir es que SMTP es mucho más viejo que HTTP: SMTP nació en 1982 y HTTP en el 1999. Precisamente, HTTP se creó, entre otros motivos, porque SMTP no era suficientemente potente (más abajo hay un apartado con las diferencias). ¿Y por qué ahora el correo electrónico no se hace a través de HTTP? Pues en parte, sí que se hace (luego desarrollaré máaaas), pero la cosa es que SMTP se ha seguido actualizando con el tiempo y lo utilizamos para mantener la compatibilidad con los sistemas de correo más antiguos.

### 2.1. Un ejemplo

Supongamos que Alice quiere mandar un correo a Bob.





1. Alice redacta un nuevo correo, indicando la dirección del destinatario: bob@putarenfe.es, y pulsa “Enviar”.
2. La aplicación envía el correo al servidor. Se almacena en la **cola de los mensajes pendientes por enviar**.
3. El servidor saca de la cola el mensaje, consulta el destinatario y crea una **conexión TCP con el servidor** putarenfe.es.
4. Utilizando el protocolo SMTP (abajo hay un ejemplo real), se envía el correo.
5. putarenfe.es recibe el correo en su **cola de mensajes pendientes por distribuir**. Una vez lo saca de la cola, ve que es para Bob y lo coloca en su bandeja de entrada.
6. Bob se conecta al servidor y ve que tiene un correo nuevo.

## 2.2. Funcionamiento del protocolo

De la misma manera que hemos hecho peticiones HTTP, vamos a mandar un correo desde la terminal.

Primero, me he instalado (en Linux, duh) Postfix, un programa que te monta automáticamente un servidor de correo SMTP en tu ordenador. Por si lo conoces, es lo que Apache para HTTP. De esta manera, puedo enviarme correos electrónicos en localhost puerto 25.

Conectándome con netcat ( `nc localhost 25` ), puedo enviar un correo electrónico:

```
S: 220 PCuni ESMTP Postfix (Ubuntu)
C: HELO direccionemisor.es
S: 250 PCuni
C: MAIL FROM: <emisor@direccionemisor.es>
S: 250 2.1.0 Ok
C: RCPT TO: <noexiste@PCuni>
S: 550 5.1.1 <noexiste@PCuni>: Recipient address rejected: User unknown
  in local recipient table
C: RCPT TO: <pruser@PCuni>
S: 250 2.1.5 Ok
C: DATA
S: 354 End data with <CR><LF>.<CR><LF>
C: Puta renfe.
  Si vieres lavaras
  eeeeeeeeeeeestas en la vara
  .
S: 250 2.0.0 Ok: queued as EB2F520E
C: QUIT
S: 221 2.0.0 Bye
```

# Hazte cliente de BBVA y ... ahórrate 6 meses de suscripción

WUOLAH  
+ BBVA

NETFLIX

Spotify

HBOmax

Disney+

PlayStation.Plus

DAZN

Ahora, si te abres una Cuenta Online en BBVA, te reembolsamos una de estas suscripciones durante 6 meses (hasta 9,99€/mes) al pagarla con tu tarjeta Aqua Débito

Promoción solo para nuevos clientes de BBVA. Válida hasta el 30/06/2023. Estas empresas no colaboran en la promoción.

1/6

Este número es indicativo del riesgo del producto, siendo 1/6 indicativo de menor riesgo y 6/6 de mayor riesgo.

BBVA está adherido al Fondo de Garantía de Depósitos de Entidades de Crédito de España. La cantidad máxima garantizada es de 100.000 euros por la totalidad de los depósitos constituidos en BBVA por persona.



Mi usuario de Ubuntu se llama **pruser** y el equipo se llama **PCuni**.

Vamos a entender qué ha pasado. Primero, para estos apuntes he empezado las líneas con **S:** y **C:**, para indicar si el mensaje lo ha enviado el servidor (**S:**) o yo (**C:**). Veamos línea a línea lo que pasa.

```
S: 220 PCuni ESMTP Postfix (Ubuntu)
```

Los mensajes del servidor siempre siguen esta sintaxis: **[Código de respuesta] [Texto explicando el código]**.

**220** en SMTP significa que la conexión ha sido establecida y el servidor está preparado. Este código siempre viene acompañado de la dirección del servidor (en este caso, **PCuni**).

```
C: HELO putarenfe.es
S: 250 PCuni
```

Los mensajes del cliente siempre siguen esta sintaxis: **[Comando] ([Argumento])**. La lista de comandos la puedes buscar, pero no es muy relevante para el tema.

**HELO** se utiliza para indicar al servidor que vamos a enviar un correo. Es una manera de contestar a su "saludo". En este comando se indica la dirección IP del servidor al que se conectó el emisor.

**250** nos indica que todo está OK por parte del servidor.

```
C: MAIL FROM: <emisor@direccionemisor.es>      # Indica la dirección del emisor.
S: 250 2.1.0 Ok                                  # Todo sigue OK en el servidor.
C: RCPT TO: <noexiste@PCuni>
S: 550 5.1.1 <noexiste@PCuni>: Recipient address rejected: User unknown
  in local recipient table
C: RCPT TO: <pruser@PCuni>                        # Mi cuenta personal.
S: 250 2.1.5 Ok                                  # Ahora me da el OK.
```



Abre tu cuenta



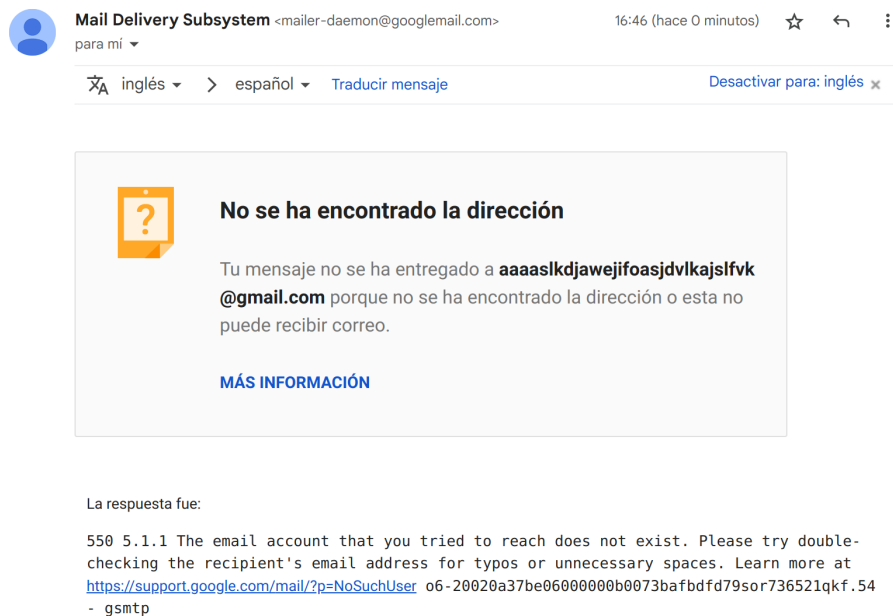
WUOLAH  
+ BBVA

**RCPT TO:** se utiliza para indicar el destinatario (*recipient*). Aquí, el servidor va a comprobar si existe esa cuenta en su sistema. Por esto, el servidor nos envía un error.

Si te fijas, en la dirección del destinatario no he tenido que poner “.com” ni “.es” ni nada. Esto se entiende mejor en el siguiente tema de DNS. Por ahora, puedes quedarte con que, como me lo estoy mandando a mi mismo ordenador, PCuni es otra manera de poner ‘localhost’.



Este sería el paso en el que Gmail nos manda un error por no encontrar la cuenta.



```
C: DATA
S: 354 End data with <CR><LF>.<CR><LF>
```

**DATA** indica al servidor que vamos a escribir el contenido del correo electrónico.

**354** indica que todo OK y que empezamos a escribir el mensaje. Además, nos indica que para señalar el final del correo lo hagamos con un salto de línea seguido de un punto seguido de otro salto de línea. Es decir, que dejemos una línea con un punto solo.



```
C: Puta renfe.  
  Si vieres lavaras  
  eeeeeeeeeeeestas en la vara  
.  
S: 250 2.0.0 Ok: queued as EB2F520E      # Recibido y todo chachi
```

Escribo el correo electrónico. Como se ve, el punto de la primera línea no hace que se envíe el correo, pero sí el de la última.

Al principio del mensaje puedes utilizar `From: [dirección del emisor]`, `To: [dirección del destinatario]` o `Subject: [Asunto del correo]` y el servidor del destinatario podrá reconocer estas directivas (para, por ejemplo, que el destinatario vea el asunto en grande y bonito).

```
C: QUIT  
S: 221 2.0.0 Bye
```

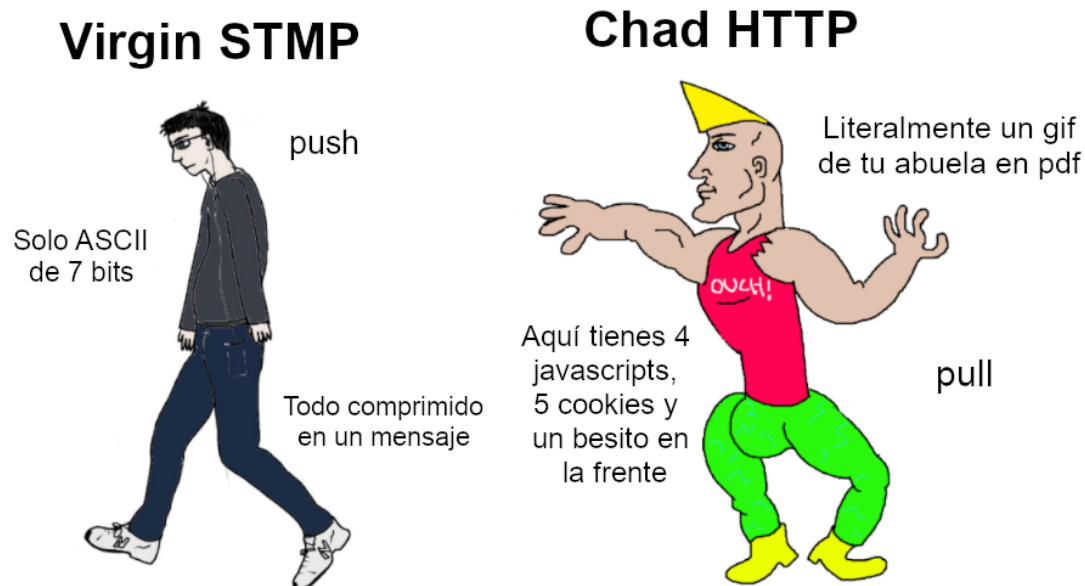
Con esto nos desconectamos del servidor, pero podríamos haber escrito otros correos poniendo el comando `MAIL FROM:`.

**Vamos a leer el correo.** Por como funciona Postfix en Ubuntu, los correos se guardan en `/var/mail/[usuario]`. Si hago `cat /var/mail/pruser`, veo lo siguiente:

```
From emisor@direccionemisor.es Mon Feb 20 17:25:25 2023  
Return-Path: <emisor@direccionemisor.es>  
X-Original-To: pruser@PCuni  
Delivered-To: pruser@PCuni  
Received: from direccionemisor.es (localhost [127.0.0.1])  
  by PCuni (Postfix) with SMTP id C9CFD20E  
  for <pruser@PCuni>; Mon, 20 Feb 2023 17:24:50 +0100 (CET)  
Message-Id: <20230220162501.C9CFD20E@PCuni>  
Date: Mon, 20 Feb 2023 17:24:50 +0100 (CET)  
From: emisor@direccionemisor.es  
  
Put a renfe.  
Si vieres lavaras  
eeeeeeeeeeestas en la vara
```

El servidor (en este caso, mi propio ordenador), ha recibido el correo.

## 2.3. Comparación con HTTP



- Desde la perspectiva del cliente, **SMTP es un protocolo push** (solo te conectas para enviar algo al servidor) y **HTTP es pull** (te conectas para pedir información al servidor).
- **SMTP solo maneja lenguaje ASCII de 7 bits.** Es decir, cuando quieres enviar algo que no es ASCII, como tildes o imágenes, el servidor emisor lo codifica a ASCII y el servidor receptor lo decodifica.
- **SMTP tiene que enviar todo el contenido comprimido en un mensaje.** En HTTP cada objeto (imágenes, css, html, ...) se envía en una respuesta separada.

## 3. Protocolos de acceso al correo

Como hemos visto, SMTP nos permite mandar y recibir correos que incluso podrían tener asunto o imágenes. Sin embargo, a la que rascamos un poco, vemos que realmente así no es como funcionan los sistemas modernos, como Gmail o Outlook.

Por ejemplo, ¿qué pasa si alguien me escribe un correo a `pruser@PCuni`, pero tengo el ordenador apagado? El servidor no estaría funcionando y **no se podría enviar el correo**. Pero, ¡hoy en día no pasa eso! Aunque tire por la ventana mi ordenador,

# Hazte cliente de BBVA y ... ahórrate 6 meses de suscripción

WUOLAH  
+ BBVA

NETFLIX

Spotify

HBOmax

Disney+

PlayStation.Plus

DAZN

Ahora, si te abres una Cuenta Online en BBVA, te reembolsamos una de estas suscripciones durante 6 meses (hasta 9,99€/mes) al pagarla con tu tarjeta Aqua Débito

Promoción solo  
para nuevos  
clientes de  
BBVA. Válida  
hasta el  
30/06/2023.  
Estas empresas  
no colaboran  
en la  
promoción.

1/6

Este número es  
indicativo del  
riesgo del  
producto, siendo  
1/6 indicativo de  
menor riesgo y  
6/6 de mayor  
riesgo.

BBVA está  
adherido al  
Fondo de  
Garantía de  
Depósitos de  
Entidades de  
Crédito de  
España.  
La cantidad  
máxima  
garantizada es  
de 100.000 euros  
por la totalidad  
de los depósitos  
constituidos  
en BBVA por  
persona.

móvil, smart watch y televisión y tire del cable del router, a la que abra la aplicación de Gmail, va a aparecer cualquier correo que me hayan enviado en ese rato.

Esto es por un concepto clave: hoy en día **el usuario está separado de su servidor de correo**. Antes, para leer y escribir correos tenías que acceder al servidor como hemos hecho en el ejemplo de más arriba. Rápidamente se dieron cuenta de que esto era un coñazo. Así, surgen los primeros protocolos de acceso al correo: **POP e IMAP**.

Estos protocolos se centran en gestionar la bandeja de entrada y pedir al servidor los correos recibidos; es decir, son **protocolos pull**. Por esto, **HTTP** también se utiliza mucho como protocolo de acceso al correo.

## 3.1. POP: puerto 110

POP1 nace en octubre de 1984 y destaca por lo sencillo que es. Hoy en día, desde 2008 (fue hace 15 años...) para ser exactos, tenemos POP3. Junto con IMAP, es un protocolo muy extendido en los servidores de correo. Sin embargo, su simpleza conlleva una funcionalidad limitada.

Veamos un ejemplo con Dovecot, un paquete de linux encargado de preparar al servidor de correo de tu ordenador (el que hemos instalado en el ejemplo de SMTP) para peticiones POP3 e IMAP.



Después de instalarlo e intentar el ejemplo que voy a poner, el servidor me contestaba con el error "ERR [SYS/PERM] Mailbox isn't a valid mbox file". Lo conseguí solucionar eliminando el salto de línea al principio del fichero `/var/spool/mail/[usuario]`.

Una vez instalado, me conecto con `nc localhost 110`.

```
S: +OK Dovecot (Ubuntu) ready.
# AUTENTICACIÓN
C: user pruser
S: +OK
C: pass [tu contraseña]
S: +OK Logged in.
# TRANSACCIÓN
C: LIST                # Lista los mensajes que tengo con su longitud
S: +OK 1 messages:
    1 448              # Mensaje 1: 448 octetos de longitud.
    .
C: RETR 1              # Petición para que nos envíe el correo 1
```

```

S: +OK 448 octets
Return-Path: <emisor@direccionemisor.es>
X-Original-To: pruser@PCuni
Delivered-To: pruser@PCuni
Received: from direccionemisor.es (localhost [127.0.0.1])
    by PCuni (Postfix) with SMTP id C9CFD20E
    for <pruser@PCuni>; Mon, 20 Feb 2023 17:24:50 +0100 (CET)
Message-Id: <20230220162501.C9CFD20E@PCuni>
Date: Mon, 20 Feb 2023 17:24:50 +0100 (CET)
From: emisor@direccionemisor.es

Put a renfe.
Si vieres lavarlas
eeeeeeeeeeeeestas en la vara
.
C: DELE 1                                # Quiero borrar el correo 1.
S: +OK Marked to be deleted.             # Lo marca como borrado, PERO NO LO BORRA AÚN.
C: RETR 1                                # Petición para el correo 1.
S: -ERR Message is deleted.              # Como está marcado como borrado, error.
C: RSET                                  # Recupera los mensajes que he marcado para borrar EN ESTA SESIÓN.
                                         # Si saliera y vuelvo a entrar, no se podría recuperar el correo 1.
S: +OK
C: RETR 1
+OK 448 octets # Volvemos a tener el correo 1.
Return-Path: <emisor@direccionemisor.es>
X-Original-To: pruser@PCuni
Delivered-To: pruser@PCuni
Received: from direccionemisor.es (localhost [127.0.0.1])
    by PCuni (Postfix) with SMTP id C9CFD20E
    for <pruser@PCuni>; Mon, 20 Feb 2023 17:24:50 +0100 (CET)
Message-Id: <20230220162501.C9CFD20E@PCuni>
Date: Mon, 20 Feb 2023 17:24:50 +0100 (CET)
From: emisor@direccionemisor.es

Put a renfe.
Si vieres lavarlas
eeeeeeeeeeeeestas en la vara
.
DELE 1                                # Volvemos a marcarlo para borrarlo.
+OK Marked to be deleted.
QUIT
# ACTUALIZACIÓN
+OK Logging out, messages deleted. # El servidor borra definitivamente el correo 1.

```

En POP, existen 3 fases:

- **Autenticación:** nos conecta al usuario dentro del servidor.
- **Transacción:** peticiones para leer correos y marcarlos para borrar. Aunque existen varios comandos que no he utilizado en el ejemplo, en esencia POP solo puede leer y borrar.
- **Actualización:** tras el usuario hace **QUIT**, el servidor borra los correos marcados

De hecho, si abro el fichero `/var/mail/pruser`, ya no está el correo, *pero hay uno random que te instala Dovecot, ni caso a ese.*

## 3.2. IMAP: puerto 143

Pero, yo en Gmail puedo hacer más cosas que leer y borrar mensajes. Por ejemplo, puedo **crear carpetas de correos** (Spam, Destacados, ...) que se **compartirán a través de mis dispositivos** y tengo un **buscador** para encontrar correos según palabras clave (*friendly reminder de que no menciono funcionalidades como enviar mensajes a varios destinatarios o programar envíos porque estos protocolos son de ACCESO a tu bandeja de entrada; es decir, para leer y gestionar los correos que te han llegado a ti*).

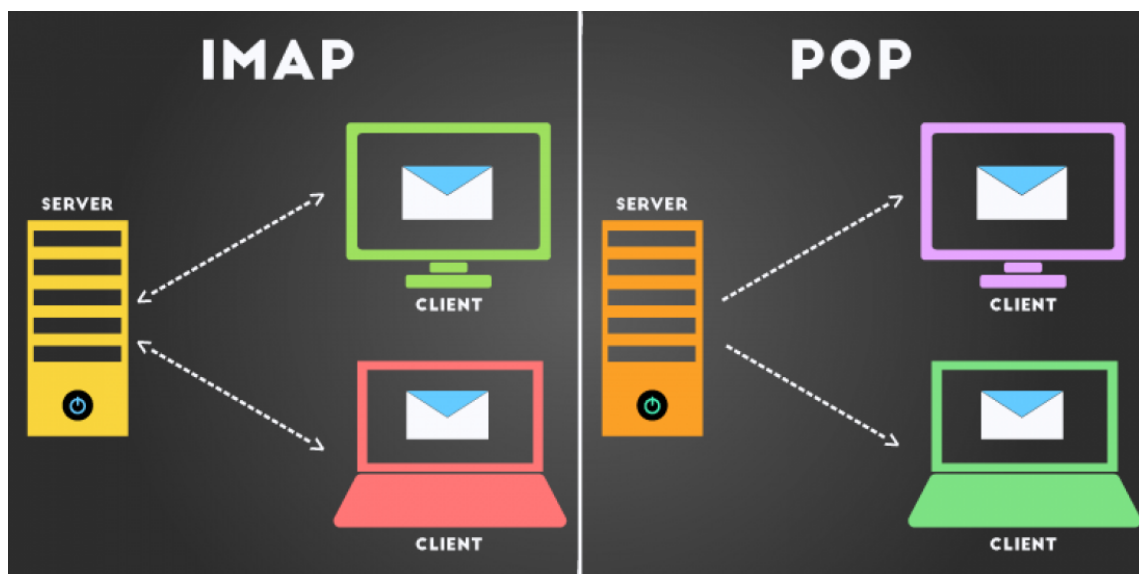
Para aportar estas características nace IMAP en 1986. Pero claro, todo esto implica más complejidad en la implementación del protocolo.



Actualmente, POP ofrece al cliente 13 comandos e IMAP, 28.

Por otro lado, las posibles respuestas de un servidor POP son 2 (OK o ERR) mientras que de IMAP pueden ser 15 distintas.

Principalmente, IMAP asocia cada mensaje a una carpeta: cuando llega un correo nuevo, se va a `INBOX` (carpeta por defecto). Ahora, IMAP te ofrece comandos para crear nuevas carpetas, renombrarlas e incluso compartirlas entre usuarios. Toda esta estructura se crea **en el lado del servidor**, por lo que aunque te conectes a tu cuenta desde las pantallas del Burger King, tus carpetas van a seguir ahí.



*La primera imagen que he encontrado para que no sea mucho texto.*

Otras funcionalidades interesantes son la posibilidad de **buscar correos** utilizando palabras clave y poder **obtener líneas específicas**, algo útil cuando trabajas con conexiones lentas y solo quieres cargar texto y no el vídeo adjunto de la comunión de la prima.

Y, ¿por qué no actualizar POP en vez de crear un protocolo nuevo? Porque la filosofía de IMAP es completamente distinta: el acceso al servidor se hará de manera **dinámica**. Mientras que POP solo se conecta al servidor para pedirle un correo y se desconecta inmediatamente, IMAP va a **permanecer conectado** desde que pases la fase de autenticación. Esto permite notificar al usuario inmediatamente al recibir un nuevo correo y, aunque te conectes simultáneamente desde diferentes dispositivos, **los cambios se cargarán en tiempo real**.

### 3.3. HTTP: puerto de indias

Y por supuesto, tenía que aparecer. Conforme la tecnología avanzó (paciencia, aún estamos hablando de la época de Internet Explorer), más gente empezó a entrar a su correo desde un **navegador web** y el tráfico de los clientes para comunicarse con sus servidores de correo pasó a ser HTTP (*otro friendly reminder: aunque los clientes usen protocolos como POP, IMAP o HTTP, los servidores de correo se comunican entre sí usando SMTP*).

De esta manera, cuando entras a tu cuenta de Gmail desde Google Chrome, mandas una petición HTTP al servidor de correo y este te contesta (con una o más



Hazte cliente de BBVA y ...

WUOLAH  
+ BBVA

**ahórrate 6 meses de suscripción**

NETFLIX

Spotify

HBOmax

Disney+

PlayStation.Plus

DAZN

Ahora, si te abres una Cuenta Online en BBVA, te reembolsamos una de estas suscripciones durante 6 meses (hasta 9,99€/mes) al pagarla con tu tarjeta Aqua Débito

Promoción solo  
para nuevos  
clientes de  
BBVA. Válida  
hasta el  
30/06/2023.  
Estas empresas  
no colaboran  
en la  
promoción.

1/6

Este número es  
indicativo del  
riesgo del  
producto, siendo  
1/6 indicativo de  
menor riesgo y  
6/6 de mayor  
riesgo.

BBVA está  
adherido al  
Fondo de  
Garantía de  
Depósitos de  
Entidades de  
Crédito de  
España.  
La cantidad  
máxima  
garantizada es  
de 100.000 euros  
por la totalidad  
de los depósitos  
constituidos  
en BBVA por  
persona.

respuestas, ventajas del HTTP) con tu bandeja de entrada. Además, cuando quieras mandar un mensaje, mandarás un mensaje HTTP con el correo electrónico y el servidor te lo mandará. Es decir, está pensado como una **app Web**.

Sin embargo, los móviles de repente se hicieron súper potentes y pasamos de entender internet como un ente detrás de nuestro navegador a descentralizarlo en muchas aplicaciones. De hecho, probablemente la mayoría de veces utilices tu correo desde la app de tu smartphone y todas estas vuelven funcionan con las últimas versiones de POP e IMAP (te animo a buscar en Wikipedia qué protocolo utiliza tu app de correo de confianza).

*Las modas siempre vuelven.*

~ Mi abuela



Abre tu cuenta



WUOLAH  
+ BBVA