

Una consulta **NS** no proporciona directamente información sobre cuál es el servidor primario y cuál es secundario. Los servidores listados en una consulta NS pueden ser tanto primarios como secundarios, y todos responden con los mismos datos debido a que replican la misma información (gracias a la transferencia de zona).

Para saber **cuál es el servidor primario**, deberías realizar una consulta **SOA** (Start of Authority), como se explicó antes. El registro SOA indica cuál es el servidor primario para esa zona.

10)**Resumen de las respuestas:**

* **Opción a**: Verdadero
* **Opción b**: Falso
* **Opción c**: Falso
* **Opción d**: Verdadero
* **Opción e**: Falso

8)

Aquí tienes las respuestas a las preguntas que aparecen en la imagen sobre el encabezado de un correo electrónico:

### **Pregunta 8.a**

* **El header de un e-mail "To:" lo agrega:**
  + La respuesta correcta es **MUA remoto** (opción **d**).
  + **Explicación**: El encabezado "To:" es generalmente agregado por el **MUA** (Mail User Agent) que está en el cliente remoto que está enviando el correo electrónico. Esto representa la dirección del destinatario al que se envía el mensaje.

### **Pregunta 8.b**

* **Con qué comando está relacionado:**
  + La respuesta correcta es **MAIL FROM** (opción **b**).
  + **Explicación**: El comando MAIL FROM se utiliza en el contexto de SMTP para especificar la dirección de correo del remitente, que es una parte crucial en la comunicación de correo electrónico.

9)El texto en la imagen pregunta sobre las **características verdaderas de TLS** (Transport Layer Security), y aquí te ofrezco un análisis de las opciones presentadas:

### **Opción a: *Implementa autenticación del servidor.***

* **Verdadero**. TLS proporciona autenticación del servidor utilizando certificados digitales. Esto asegura que el cliente esté conectado al servidor correcto, lo que es parte crucial del protocolo.

### **Opción b: *Sólo HTTP y POP3 usan TLS para asegurar las comunicaciones.***

* **Falso**. TLS no se limita solo a **HTTP** (usado en **HTTPS**) y **POP3**. También se usa en otros protocolos como **SMTP**, **IMAP**, **FTP**, entre otros, para asegurar las comunicaciones.

### **Opción c: *Provee cifrado para garantizar confidencialidad de los datos.***

* **Verdadero**. TLS ofrece cifrado de extremo a extremo para proteger la confidencialidad de los datos durante la transmisión. Esto asegura que terceros no puedan interceptar o leer la información.

### **Opción d: *Como parte del handshake el servidor envía su certificado que contiene su clave privada.***

* **Falso**. Durante el **TLS handshake**, el servidor envía su **certificado digital**, pero **no incluye la clave privada**. El certificado contiene la clave pública del servidor, que es utilizada por el cliente para cifrar la información inicial. La clave privada se mantiene segura en el servidor.

### **Opción e: *Los mensajes se cifran con la clave pública del servidor.***

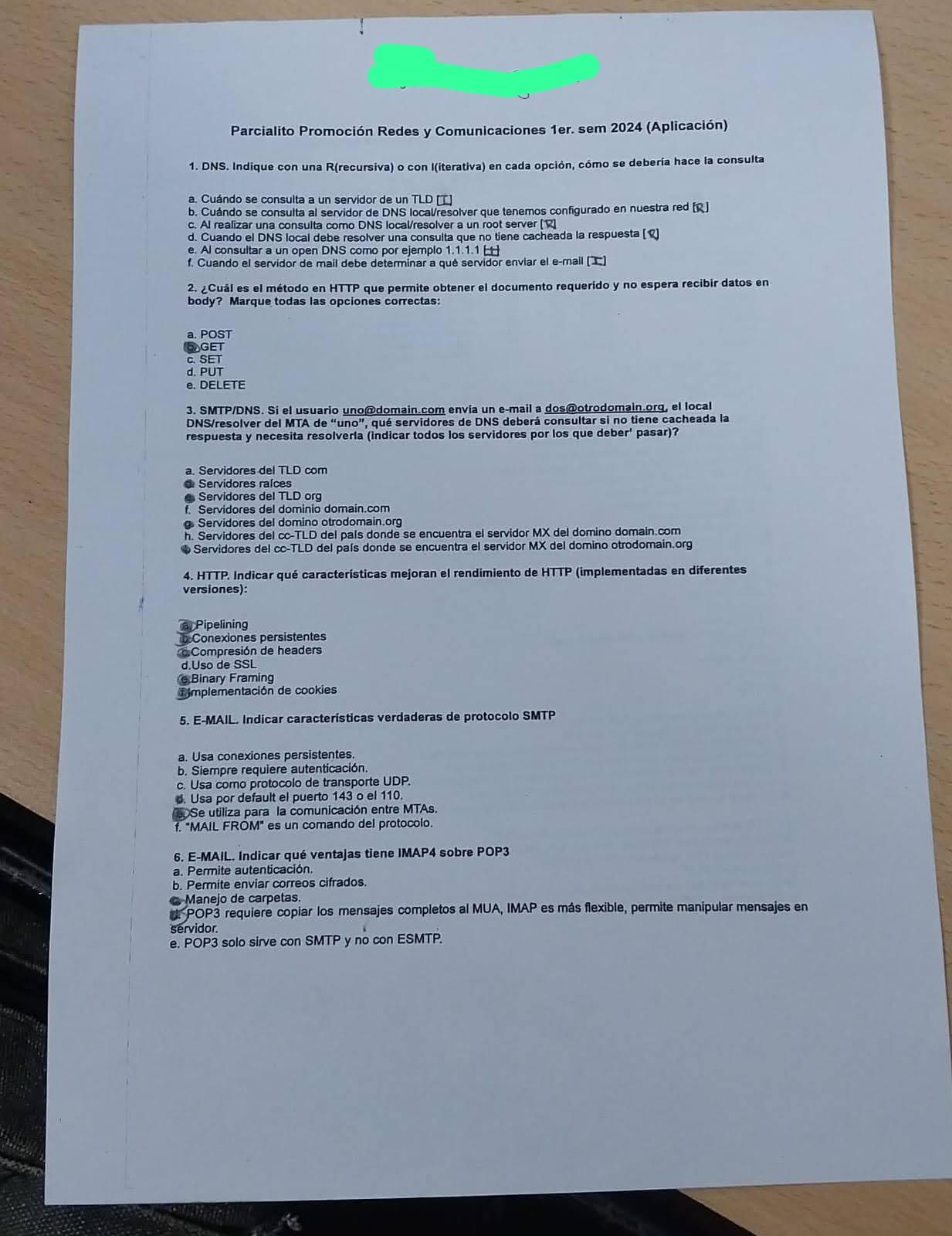
* **Verdadero, en parte**. Durante el proceso de establecimiento de una sesión TLS, inicialmente los datos se cifran utilizando la **clave pública del servidor** (para compartir un secreto o clave de sesión). Sin embargo, una vez establecida la clave de sesión, se utiliza un cifrado simétrico para la comunicación posterior.

### **Opción f: *La clave de sesión es un secreto compartido entre cliente y servidor (ambos la conocen).***

* **Verdadero**. Después del handshake, tanto el cliente como el servidor acuerdan una **clave de sesión** compartida que utilizan para cifrar y descifrar el tráfico durante la sesión. Esta clave se utiliza en un esquema de cifrado simétrico para asegurar la comunicación de ambos lados.

### **Resumen:**

* Las respuestas **a**, **c**, **e** y **f** son **verdaderas**.
* Las respuestas **b** y **d** son **falsas**.



**SMTP** usa los puertos: **25**, **465**, y **587**.

**POP3** usa el puerto: **110**.

**IMAP** usa el puerto: **143**.

**IMAP** (Internet Message Access Protocol) y **POP3** (Post Office Protocol version 3) son protocolos utilizados para la **recepción** de correos electrónicos, pero no utilizan **ESMTP** directamente. En su lugar, **ESMTP** es un protocolo para el **envío** de correos electrónicos.

La pregunta está pidiendo identificar las características que mejoran el rendimiento de HTTP en sus diferentes versiones. Vamos a repasar cada una de las opciones:

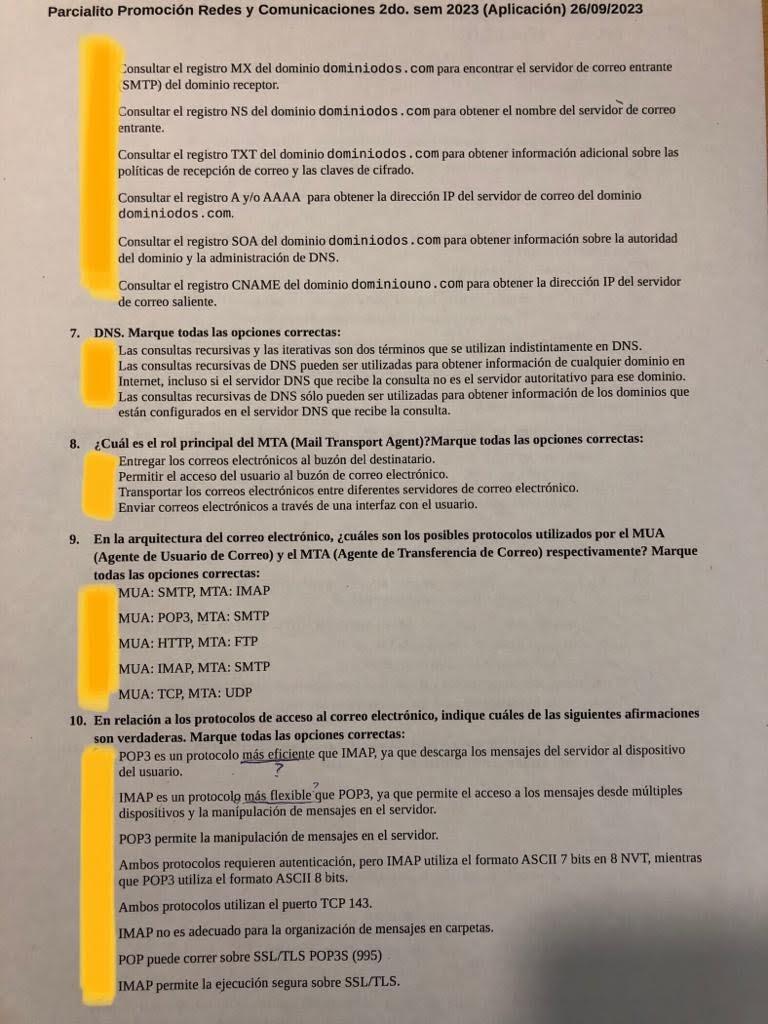
* **a. Pipelining**: El "pipelining" permite enviar múltiples solicitudes HTTP sin tener que esperar la respuesta de la anterior. Esta técnica se introdujo en HTTP/1.1 y mejora el rendimiento al reducir la latencia, aunque su adopción fue limitada debido a problemas de implementación en los navegadores.
* **b. Conexiones persistentes**: Las conexiones persistentes, también introducidas en HTTP/1.1, permiten reutilizar la misma conexión TCP para múltiples solicitudes/respuestas, en lugar de abrir una nueva conexión para cada interacción. Esto reduce la sobrecarga de conexión y mejora significativamente el rendimiento.
* **c. Compresión de headers**: Esta característica es una mejora implementada en HTTP/2. En lugar de enviar los headers completos en cada solicitud, se utilizan técnicas de compresión (como HPACK) para reducir el tamaño de los headers y mejorar el rendimiento, especialmente en conexiones con muchas solicitudes.
* **d. Uso de SSL**: El uso de SSL/TLS mejora la seguridad del protocolo, pero no necesariamente su rendimiento. De hecho, añadir cifrado puede introducir cierta sobrecarga, aunque con HTTP/2 y HTTP/3, TLS es parte esencial de las conexiones.
* **e. Binary Framing**: Introducido en HTTP/2, el "binary framing" divide los datos en pequeños fragmentos binarios, lo que permite manejar múltiples flujos de datos simultáneamente en una misma conexión. Esto mejora la eficiencia del protocolo y reduce la latencia, aumentando así el rendimiento.
* **f. Implementación de cookies**: Las cookies son utilizadas para almacenar información de la sesión entre cliente y servidor, pero no mejoran directamente el rendimiento de HTTP. Su función principal es la gestión de estado y autenticación.

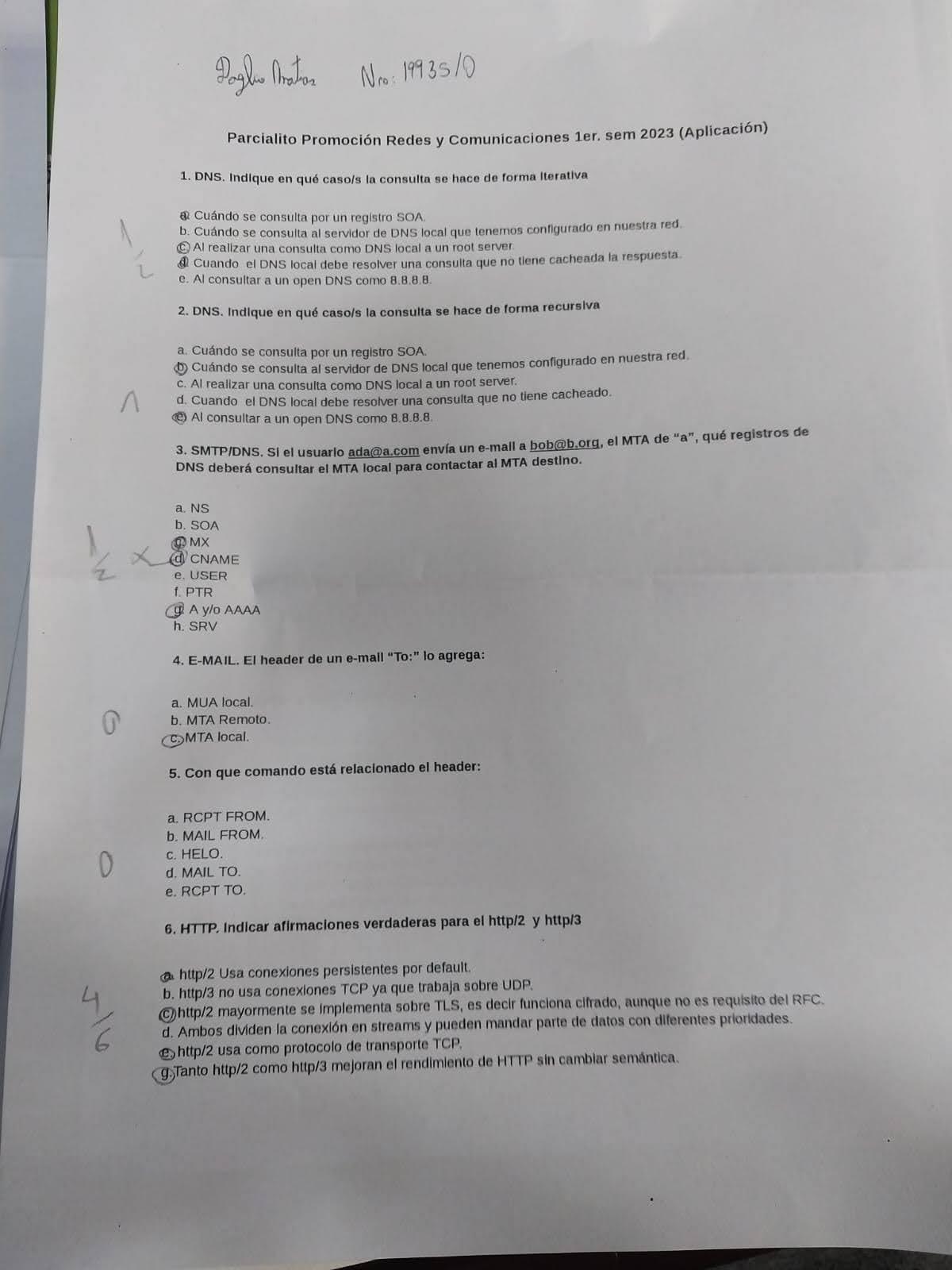
### **Respuesta final:**

Las características que mejoran el rendimiento de HTTP son:

* **a. Pipelining**
* **b. Conexiones persistentes**
* **c. Compresión de headers**
* **e. Binary Framing**

Estas características están diseñadas específicamente para reducir la latencia y mejorar el manejo de múltiples solicitudes, optimizando el rendimiento del protocolo HTTP.





Aquí tienes un desglose de cada uno de los comandos que mencionaste, junto con la razón de por qué **RCPT TO** es el único que se relaciona directamente con los headers del correo electrónico:

### **1. HELO**

* **Función**: Este comando se utiliza para iniciar una sesión SMTP. El cliente se presenta al servidor, enviando su nombre de dominio o dirección IP.
* **Relación con Headers**: No se relaciona con los headers del mensaje. Se trata más de establecer la conexión y no de definir información del correo.

### **2. MAIL FROM**

* **Función**: Este comando especifica la dirección de correo electrónico del remitente. Es el primer paso en el proceso de envío de un correo.
* **Relación con Headers**: Aunque establece quién está enviando el correo, no forma parte de los headers del mensaje. Los headers del correo incluyen información que será parte del cuerpo del mensaje y que se envía después de MAIL FROM.

### **3. RCPT TO**

* **Función**: Este comando indica al servidor la dirección de correo electrónico del destinatario.
* **Relación con Headers**: Este comando se relaciona directamente con los headers del correo porque establece a quién se envía el correo, lo cual se refleja en el header **To:** del mensaje.

### **4. DATA**

* **Función**: Este comando se utiliza para indicar que el cliente está a punto de enviar el contenido del mensaje, incluyendo los headers y el cuerpo del correo.
* **Relación con Headers**: Este comando está relacionado con los headers, ya que es en este momento cuando se definen los headers del correo (como From:, To:, Subject:) antes de enviar el cuerpo del mensaje.

### **5. RCPT FROM**

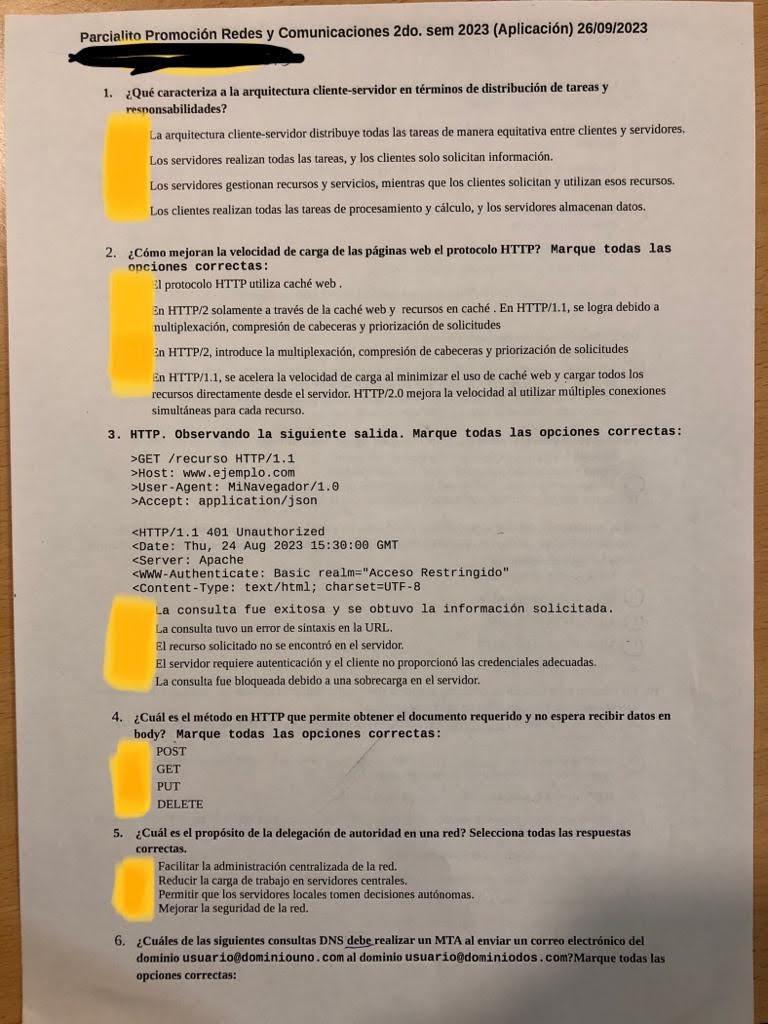
* **Función**: Este comando no es parte del protocolo SMTP estándar y no se utiliza para enviar correos. En su lugar, se utiliza MAIL FROM para especificar el remitente.
* **Relación con Headers**: Dado que no es un comando estándar, no tiene relación con los headers.

### **6. MAIL TO**

* **Función**: Este comando tampoco es parte del protocolo SMTP. El correcto es RCPT TO.
* **Relación con Headers**: Al no ser un comando válido en SMTP, no tiene relación con los headers del correo.

### **Resumen:**

* **RCPT TO** es el único comando en tu lista que se utiliza para definir el destinatario del correo, directamente relacionado con el header **To:**.
* Los comandos **HELO**, **MAIL FROM**, y **DATA** tienen funciones importantes en el proceso SMTP, pero no están directamente relacionados con los headers que se envían en el mensaje.
* **RCPT FROM** y **MAIL TO** no son comandos válidos en SMTP, por lo que tampoco tienen relación con los headers.

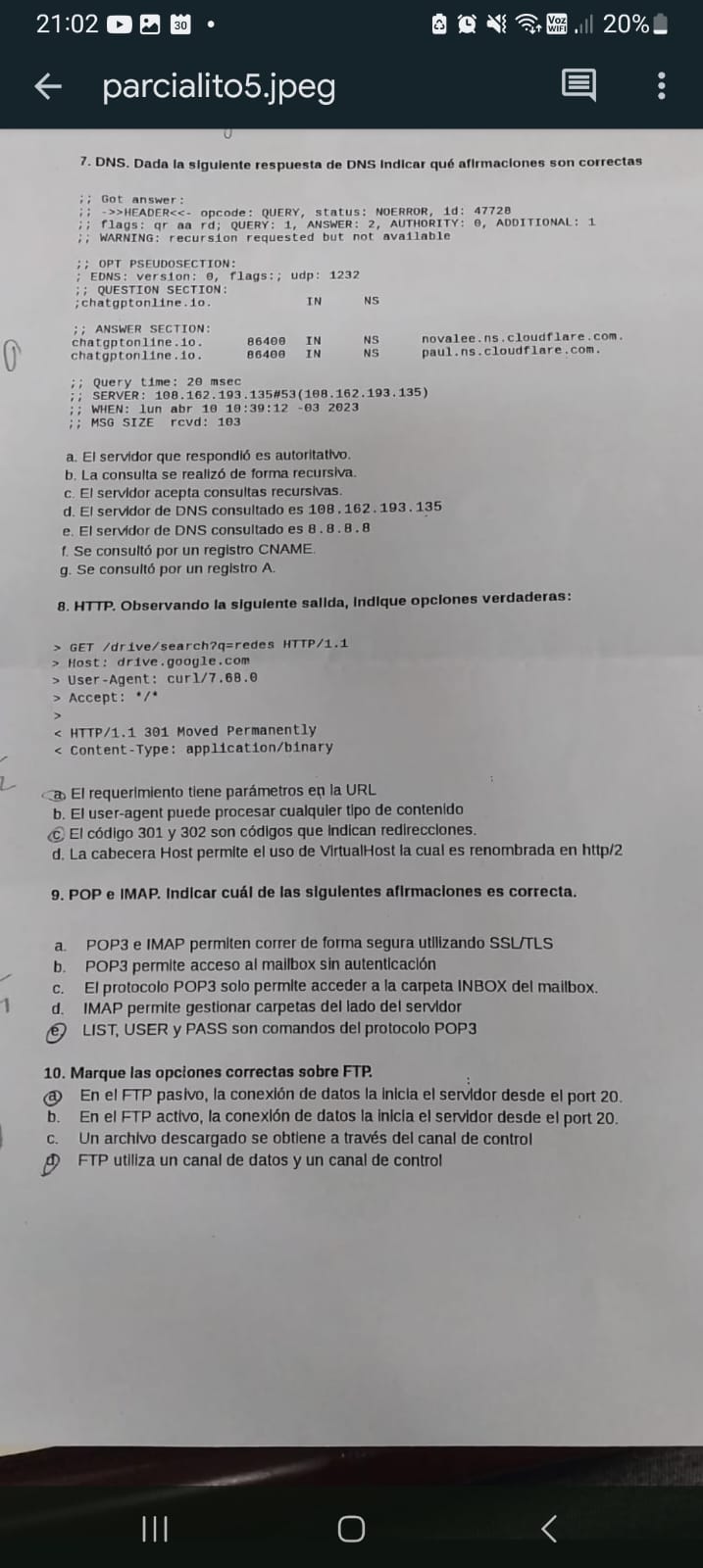


¿Cuáles son las diferencias?

Una de las principales diferencias entre los procesos del lado del servidor y del lado del cliente es la cantidad de control y acceso a los recursos que tiene cada uno. Los procesos del lado del servidor tienen acceso a los recursos del servidor, como su CPU, memoria y almacenamiento, así como a cualquier base de datos u otros servidores que utilice la aplicación web. Los procesos del lado del cliente, por otro lado, tienen acceso sólo a los recursos del dispositivo del usuario, como su CPU, memoria y almacenamiento.

Otra diferencia entre los procesos del lado del servidor y del lado del cliente es el nivel de confianza que se puede depositar en ellos. Debido a que los procesos del lado del servidor se ejecutan en el servidor web, normalmente son más seguros y menos vulnerables a manipulaciones o ataques maliciosos. Los procesos del lado del cliente, por otro lado, se ejecutan en el dispositivo del usuario, lo que significa que son potencialmente menos seguros y más susceptibles a manipulaciones o ataques.

En resumen, el lado del servidor y el lado del cliente se refieren a la ubicación donde se llevan a cabo ciertas tareas o procesos en una aplicación web. Los procesos del lado del servidor se ejecutan en el servidor web, mientras que los procesos del lado del cliente se ejecutan en el dispositivo del usuario. Estos procesos tienen diferentes niveles de acceso a los recursos y confianza, y desempeñan diferentes roles en el funcionamiento general de una aplicación web.



### **Sobre POP e IMAP:**

1. **a)** POP3 e IMAP permiten correr de forma segura utilizando SSL/TLS.
   * **Verdadero**: Ambos protocolos pueden operar sobre SSL/TLS para asegurar la conexión.
2. **b)** POP3 permite acceso al mailbox sin autenticación.
   * **Falso**: POP3 requiere autenticación para acceder al correo.
3. **c)** El protocolo POP3 solo permite acceder a la carpeta INBOX del mailbox.
   * **Verdadero**: POP3 generalmente solo da acceso a la bandeja de entrada (INBOX).
4. **d)** IMAP permite gestionar carpetas del lado del servidor.
   * **Verdadero**: IMAP permite gestionar y sincronizar carpetas en el servidor.
5. **e)** LIST, USER y PASS son comandos del protocolo POP3.
   * **Falso**: LIST es un comando de IMAP, mientras que USER y PASS son de autenticación en ambos.

### **Sobre FTP:**

1. **a)** En el FTP pasivo, la conexión de datos la inicia el servidor desde el puerto 20.
   * **Falso**: En el modo pasivo, el cliente inicia la conexión de datos, no el servidor.
2. **b)** En el FTP activo, la conexión de datos la inicia el servidor desde el puerto 20.
   * **Verdadero**: En el modo activo, el servidor se conecta al cliente en el puerto designado por este.
3. **c)** Un archivo descargado se obtiene a través del canal de datos.
   * **Verdadero**: La transferencia de archivos se realiza a través del canal de datos.
4. **d)** FTP utiliza un canal de datos y un canal de control.
   * **Verdadero**: FTP utiliza dos canales, uno para el control de la conexión y otro para la transferencia de datos.