Tarea 6

MÉTODO PUT EN HTTP

FECHA

16 de junio de 2025

MATERIA

Aplicaciones para comunicaciones de red

PROFESOR

Moreno Cervantes Axel Ernesto

ALUMNOS

- Guevara Badillo Areli Alejandra
- Ramírez Martínez Alejandro

GRUPO

6CM2

Tamaño soportado y límites reales

Aunque el protocolo HTTP (según las RFC 7230 y 7231) **no impone un tamaño máximo fijo** para el contenido enviado mediante PUT, en la práctica **sí existen límites reales** impuestos por servidores, frameworks, clientes y servicios intermedios.

El cuerpo de la petición se gestiona usando las cabeceras Content-Length o Transfer-Encoding, lo que permite enviar datos muy grandes **siempre que el servidor y los componentes intermedios lo permitan**.

Factores que afectan el tamaño soportado por PUT

Componente	Límite por defecto	Cómo se configura
1. Servidores Web		
Nginx	1 MB	client_max_body_size 100M; en el bloque http o server
Apache	Sin límite (valor por defecto: 0)	LimitRequestBody 104857600 (100 MB en bytes)
Tomcat	2 MB (por maxPostSize)	En el archivo server.xml del conector HTTP
ModSecurity	128 MB	SecRequestBodyLimit en configuración del módulo
2. Frameworks / Lenguajes Backend		
Node.js (Express)	100 KB (por express.json())	app.use(express.json({ limit: '50mb' }))
PHP	2 MB (upload_max_filesize)	En php.ini: upload_max_filesize y post_max_size

ASP.NET / IIS	30 MB	En web.config: maxRequestLength, maxAllowedContentLength
Java (Jetty, JBoss)	Varía según contenedor	Usan parámetros similares: maxPostSize, maxFormContentSize, etc.
3. Clientes HTTP / Navegadores		
Curl, Postman, navegadores	No imponen límites estrictos	Limitados por RAM disponible o configuración del sistema operativo
4. Proxies y Firewalls		
ModSecurity	128 MB	Mismo parámetro: SecRequestBodyLimit
Amazon API Gateway	10 MB	No configurable en planes estándar
Cloudflare	100 MB	Límite global por solicitud HTTP (no configurable en todos los planes)
Squid / Apache Traffic Server	1 a 10 MB según configuración	Se puede ajustar en sus archivos de configuración

¿Qué pasa si se supera el límite?

Cuando el cuerpo de la petición PUT sobrepasa alguno de estos límites:

- **Se retorna un error HTTP 413**: Request Entity Too Large.
- También pueden aparecer errores 502 o 504 si el intermediario (como un proxy) cancela la conexión.
- Algunos sistemas pueden rechazar la solicitud sin respuesta clara si no tienen manejo adecuado de errores.

Conclusión

Aunque el protocolo HTTP permite técnicamente enviar cuerpos de cualquier tamaño mediante el método PUT, en la realidad esto está condicionado por las configuraciones de los distintos componentes del sistema, como servidores web, frameworks backend, proxies y herramientas de seguridad. Cada uno de ellos puede imponer su propio límite al tamaño de la solicitud.

Por ello, al implementar servicios que requieran recibir datos grandes —como cargas de archivos o actualizaciones masivas— es importante revisar y modificar estos límites según las necesidades del proyecto. Hacerlo evita errores como el **HTTP 413 (Request Entity Too Large)** y mejora la estabilidad y fiabilidad del sistema en producción.