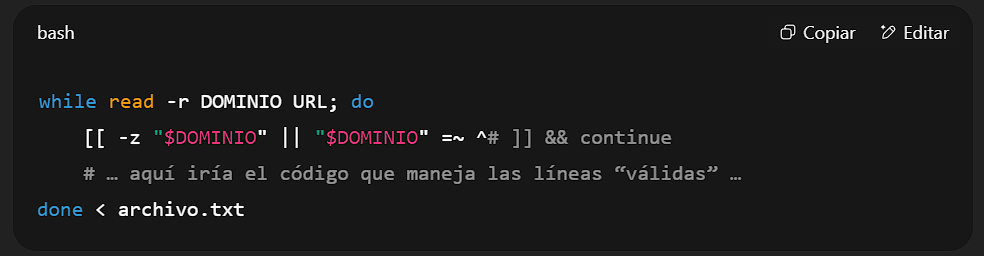
EXPLICACION DE SCRIPT DE URL\_CHECK.

  
Lee cada línea de archivo.txt, separando en dos variables:

DOMINIO ← primer “token”

URL ← segundo “token” (o el resto de la línea)

Si la línea está vacía o empieza con #, no procesa nada y pasa a la siguiente.

En caso contrario, sigue con el código que pongas debajo (por ejemplo, hacer un curl $URL o lo que necesites).



[[ ! "$URL" =~ ^https?:// ]]

Usa la extensión de Bash =~ para evaluar una expresión regular.

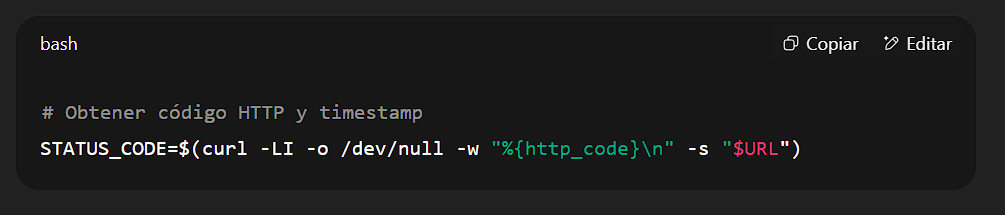
^https?:// coincide con http:// o https:// al principio de la cadena.

El ! niega la condición, de modo que la prueba es “si no empieza por http:// o https://…”.

&& URL="http://$URL"

Si la prueba da true (no tiene protocolo), entonces antepone http:// a la variable.

Ejemplo: si URL="example.com", tras esto URL pasará a ser "http://example.com".



curl con estas opciones:

-L sigue redircciones (301, 302…).

-I pide solo el HEAD (cabeceras), no el cuerpo.

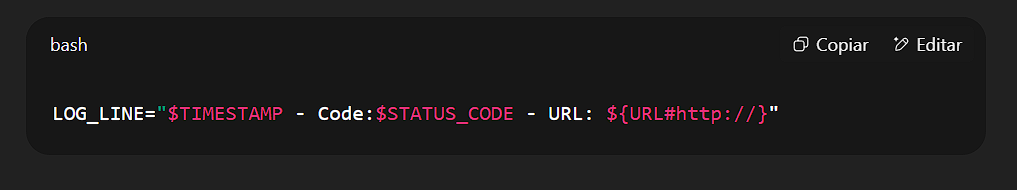
-o /dev/null descarta cualquier salida en stdout.

-w "%{http\_code}\n" imprime únicamente el código de estado HTTP (p. ej. 200, 404).

-s modo silencioso, oculta la barra de progreso y mensajes de error.

Todo eso se captura con $(…) en la variable STATUS\_CODE.

Al final STATUS\_CODE contendrá algo como 200.



Construye una cadena para el log, con tres partes separadas por -:

El timestamp (20250703\_174523)

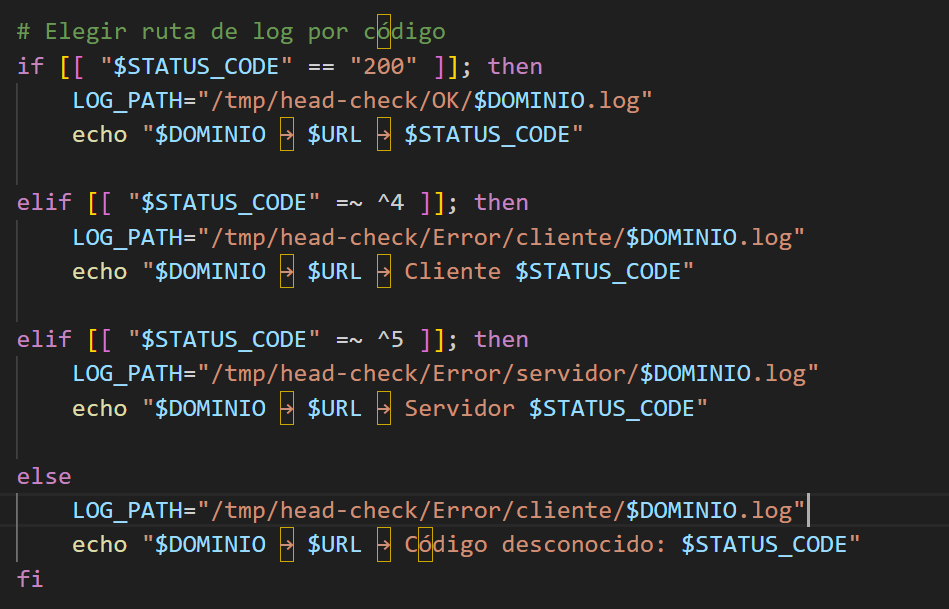
El código HTTP (Code:200)

La URL, pero sin el prefijo http:// (o https://):

${URL#http://} usa expansión de parámetros de Bash:

Quita el primer prefijo http:// si existe.

Si fuera https://, ese prefijo no lo quita, así que podrías usar ${URL#\*://} para eliminar ambos por igual.

  
  
  
Desglose de cada tramo

if [[ "$STATUS\_CODE" == "200" ]]; then

Verifica igualdad exacta con 200.

elif [[ "$STATUS\_CODE" =~ ^4 ]];

La expresión regular ^4 comprueba si comienza por “4” (cualquier 4xx).

=~ es el operador de regex en Bash.

elif [[ "$STATUS\_CODE" =~ ^5 ]];

Análogo al anterior pero para 5xx (servidor).

En cada rama:

Se fija LOG\_PATH en la carpeta correspondiente (OK, cliente o servidor).

Se usa echo para imprimir un mensaje legible que indique dominio, URL y tipo de respuesta.

El else final cubre cualquier otro caso (redirecciones, informativos…) y lo clasifica por defecto como “error de cliente”.