

Sockets con Python

Servidor HTTP

Autor: González, Agustín

Descripción: Implementación de **servidor HTTP** básico, que da respuesta a solicitudes del tipo **GET**, vía los códigos **200** (OK), **404** (Error) y **304** (Not Modified).

Ejecución: Para ejecutar el script `“http_server_run.py”` se utiliza, por defecto, el archivo de configuración `“include/config/ http_server_run.conf”`, del que es posible modificar los siguientes parámetros:

- **Host:** Dirección IP (anfitrión), por defecto `127.0.0.1`.
- **Port:** Puerto, por defecto `8080`.
- **Webdir:** Directorio web, por defecto `“include/data/www”`.
- **Idxpath:** Path del archivo índice relativo al directorio web. Por defecto `index.html`.
- **Errpath:** Path del archivo de error 404 (recurso no encontrado), también relativo a Webdir. `404.html` por defecto.
- **Logpath:** Path del log (`“output/http_server_run/server_log.txt”`).

Ejemplo de ejecución:

```
> python http_server_run.py
```

Funcionamiento: Los siguientes encabezados se envían junto a la respuesta:

- **Content-Length:** Tamaño de recurso.
- **Cache-Control:** Máximo tiempo en caché de un recurso (seteado como max-age: 604800).
- **Content-Type:** Tipo de contenido.

En caso de una petición condicional, el servidor retorna el código 304, si el recurso no ha sido modificado desde la fecha indicada por el cliente.

Desarrollo: La implementación de la clase **BaseHTTPServer** (`“include/lib/”`) es muy similar a la de Python, del mismo nombre, aunque con menores funcionalidades, a efectos de la resolución del ejercicio original del trabajo práctico. Esta clase base, hereda su funcionalidad a la clase **Server**, la cual rescribe el método **do_GET()**, llevado a cabo ante una petición de ese tipo.