

Relazione Tecnica - Web Server Python + Sito Web Statico

Nome: Pietro Siboni

Matricola: 0001118700

Corso: Programmazione di Reti - Laboratorio

Titolo

Realizzazione di un Web Server minimale in Python e pubblicazione di un sito statico.

Obiettivo

L'obiettivo del progetto è stato quello di creare un semplice web server in Python utilizzando il modulo `socket`, capace di gestire richieste HTTP di tipo GET e servire pagine HTML statiche da una cartella locale (`www/`).

Struttura del Progetto

- `Server.py` : codice del server Python (basato su socket TCP).
- `www/` : contiene i contenuti statici:
 - `home.html` → pagina principale del sito.
 - `about.html` → sezione informativa.
 - `contact.html` → informazioni di contatto.
 - `style.css` → foglio di stile con layout responsive e animazioni.
 - `fotoTraccia.jpg` → immagine visualizzabile nel sito.

Funzionalità Implementate

Requisiti Minimi

- Il server risponde su *localhost:8080*.
- Gestione di richieste HTTP GET.
- Risposta *200 OK* per file esistenti.
- Risposta *404 Not Found* per file inesistenti.
- Servizio di almeno tre pagine HTML statiche.

Estensioni Opzionali Implementate

- Gestione automatica dei MIME types (*.html*, *.css*, *.jpg*).
- Logging delle richieste.
- Aggiunta di animazioni o layout responsive.

Descrizione Tecnica

Server Python (*Server.py*)

- Usa *socket.AF_INET* e *SOCK_STREAM* → connessione TCP.
- Riceve le richieste HTTP, estrae il metodo e il percorso richiesto.
- Cerca i file nella cartella *www/* usando *os.path.join()*.
- Usa *mimetypes.guess_type()* per inviare il giusto *Content-Type*.
- Logga ogni richiesta con *print()* e *timestamp*.
- Risponde con *200 OK* o *404 Not Found* con data e ora.

Contenuti Statici (*www/*)

- Tre pagine HTML collegate tra loro tramite link.
- *style.css* gestisce lo stile per le tre pagine HTML.
- Immagine *fotoTraccia.jpg* che rappresenta una foto della traccia data.

Esecuzione del Progetto

1. Aprire *Server.py* in un ambiente Python.
2. Avviare lo script: verrà avviato su *http://localhost:8080*.
3. Aprire un browser e visitare le pagine:
 - *http://localhost:8080/home.html*
 - *http://localhost:8080/about.html*
 - *http://localhost:8080/contact.html*
 - *http://localhost:8080/style.css*
 - *http://localhost:8080/fotoTraccia.jpg*
4. Provare anche una pagina inesistente per testare la risposta *404* (*http://localhost:8080/hello.html*).

Conclusione

Il progetto rispetta tutti i requisiti minimi previsti dalla traccia, implementando un web server TCP in Python, in grado di servire contenuti statici. Inoltre, ha incluso estensioni grafiche e funzionali opzionali come gestione MIME, logging e stile responsive.