

Relazione Progetto Programmazione di Reti

Sohaib Ouakani

Maggio 2025

Indice

1 Obiettivo del Progetto	1
2 Organizzazione del Progetto	1
3 Funzionalità Obbligatorie Implementate	2
4 Funzionalità Opzionali Implementate	2
5 Funzionalità Extra Implementate	2
6 Conclusione	3

1 Obiettivo del Progetto

L'obiettivo di questo progetto è la creazione di un Web server minimale in Python che possa servire un sito scritto in HTML. Il server deve gestire correttamente richieste di tipo GET e rispondere con codici di stato adeguati. Inoltre, deve fornire una risposta per richieste di risorse non esistenti. Il server risponde sull'indirizzo `localhost` alla porta 8080 e fornisce un sito composto da almeno tre pagine HTML.

2 Organizzazione del Progetto

Il progetto è composto da:

- Una cartella `www` che contiene i file relativi al sito, ovvero file `.html` e i fogli di stile `.css`;
- Il file `server.py`, che contiene il codice Python per l'implementazione del server;

- Il file `log.txt`, contenente il registro delle richieste gestite dal server. Se non presente, verrà creato automaticamente all'avvio.

3 Funzionalità Obbligatorie Implementate

Una volta avviato, il server si mette in ascolto su `localhost:8080` in attesa di richieste da parte di un client. All'arrivo di una richiesta, questa viene analizzata per verificarne l'integrità e determinare la risorsa richiesta. Successivamente, viene costruita una risposta conforme al protocollo HTTP e inviata al client. Il server torna poi in attesa di ulteriori richieste.

In caso venga richiesta una risorsa inesistente, viene fornita una pagina HTML che segnala l'errore (pagina 404). Se non viene specificata alcuna risorsa, viene inviata la pagina `index.html`, che funge da homepage del sito.

4 Funzionalità Opzionali Implementate

Il server specifica correttamente il tipo di contenuto servito, compilando il campo **Content-Type** nella risposta HTTP. È stata inoltre implementata una funzione di logging, che registra nel file `log.txt` data e ora della richiesta, indirizzo e porta del client, metodo HTTP, risorsa richiesta, codice di risposta, messaggio associato e tempo di risposta in millisecondi.

Il sito web fornito include animazioni e un layout responsive, con un risultato esteticamente gradevole e accessibile da vari dispositivi.

5 Funzionalità Extra Implementate

Tra le funzionalità extra implementate vi sono:

- Un sistema di sicurezza che impedisce l'accesso a risorse al di fuori della cartella `www`, rispondendo con un codice 403 (Forbidden);
- Supporto al multithreading per la gestione simultanea delle richieste: per ogni richiesta viene creato un thread dedicato, migliorando la scalabilità del server;
- Ottimizzazione del layout del sito per la visualizzazione su dispositivi mobili, come smartphone e tablet.

6 Conclusione

Il server web sviluppato rappresenta una soluzione minimalista ma efficace per l'hosting di contenuti statici. Il server è in grado di gestire richieste HTTP di tipo GET, fornendo risposte corrette in termini di codice di stato e contenuto, con una gestione appropriata degli errori e delle risorse non disponibili.

Le funzionalità opzionali implementate, come il logging dettagliato e il supporto al multithreading, contribuiscono a incrementare l'affidabilità e la scalabilità del server. Il layout responsive del sito garantisce un utilizzo ottimale anche da dispositivi mobili, migliorando l'esperienza utente.

Il progetto costituisce una base solida ed eventualmente estendibile. Tra i possibili sviluppi futuri si possono includere il supporto a richieste POST e l'adozione di framework web per applicazioni più complesse. Questi miglioramenti permetterebbero di evolvere il server verso un servizio web più completo e conforme agli standard moderni.