



Relazione elaborato

Programmazione di Reti

Traccia 2 - Python Web Server

Andrea Zammarchi

0000914652

Giugno 2021

Indice

1	Introduzione	3
2	Descrizione	3
3	Utilizzo	3
4	Implementazione	4
4.1	Sito Web	4
4.2	GUI login	5
4.3	Web Server	6

1 Introduzione

Questo progetto ha l'obiettivo di creare un Web Server http multithread in *Python* con login tramite credenziali. Questo Web Server è di proprietà di una clinica ospedaliera immaginaria: *Clinica Zammarchi*.

2 Descrizione

Per poter avviare il server è necessario innanzitutto effettuare l'autenticazione tramite la finestra di accesso che apparirà all'avvio del programma. Se si è nuovi basta digitare un username e una password a piacere e si verrà aggiunti all'archivio degli utenti registrati. Altrimenti si può semplicemente effettuare il login tramite username e password utilizzati in precedenza.

Nella Homepage sono elencati tutti i servizi offerti dalla clinica, ovvero:

- Fisioterapia
- Riabilitazione
- Palestra

Ogni servizio ha inoltre una pagina dedicata con ulteriori informazioni a riguardo. Nella Homepage in aggiunta è possibile scaricare un file pdf, in questo caso è il manuale dei server http per *Python*.

3 Utilizzo

Per poter eseguire l'applicazione è necessario aver installate, oltre alle librerie incluse nel pacchetto Python base, la libreria PIL. Per installare tale libreria si consiglia di utilizzare il package manager pip.

```
python3 -m pip install --upgrade pip
python3 -m pip install --upgrade Pillow
```

Per avviare il programma aprire la console, spostarsi all'interno della directory `src` ed eseguire il comando:

```
python http_multithread_server.py PORT
```

Il parametro PORT specifica la porta su cui aprire il web server, ma questa è opzionale (default 8080). Si aprirà quindi una finestra di login dove bisognerà registrarsi/accedere. Una volta eseguito l'accesso il server sarà attivo, quindi aprire il browser e fare la richiesta al seguente link: <http://127.0.0.1:PORT>. Per terminare il server basta selezionare la console e digitare **Ctrl-C**.

4 Implementazione

4.1 Sito Web

La creazione delle pagine del sito web è stata svolta con l'ausilio di un website builder, ovvero *Nicepage*. La scelta è stata dettata dall'efficienza che tale applicazione offre, in fatto di qualità e di tempistiche. Permette di generare in automatico centinaia di righe di codice HTML in pochi click.

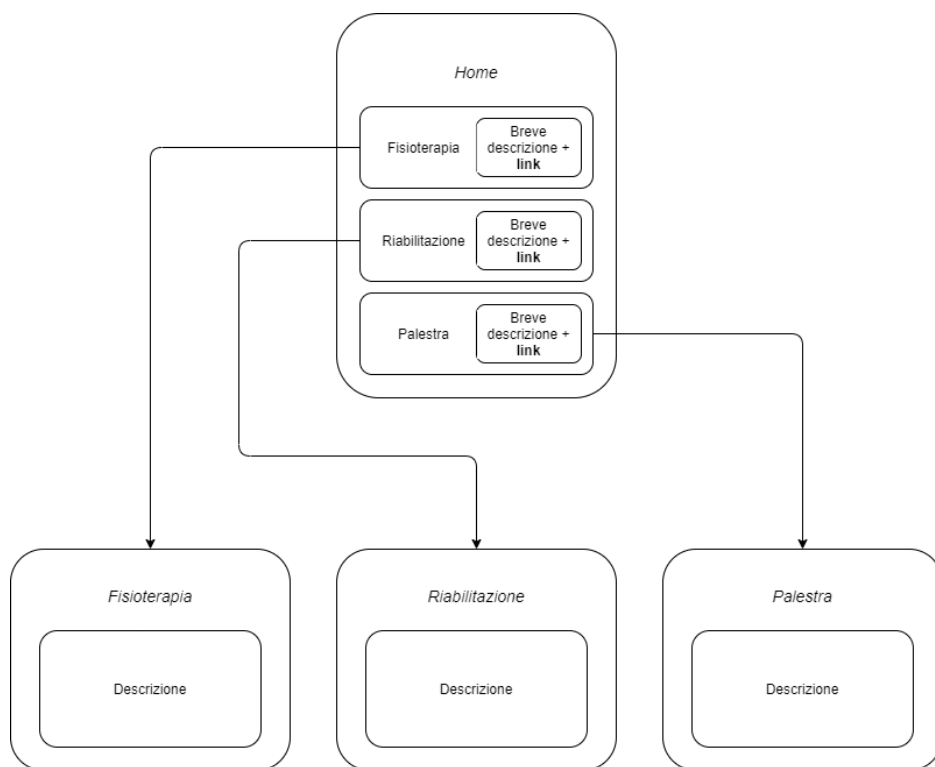


Figura 1: Struttura sito web

4.2 GUI login

Per quanto riguarda invece l'interfaccia grafica di login, è stata utilizzata la libreria inclusa nel package base di *Python*, cioè Tkinter.

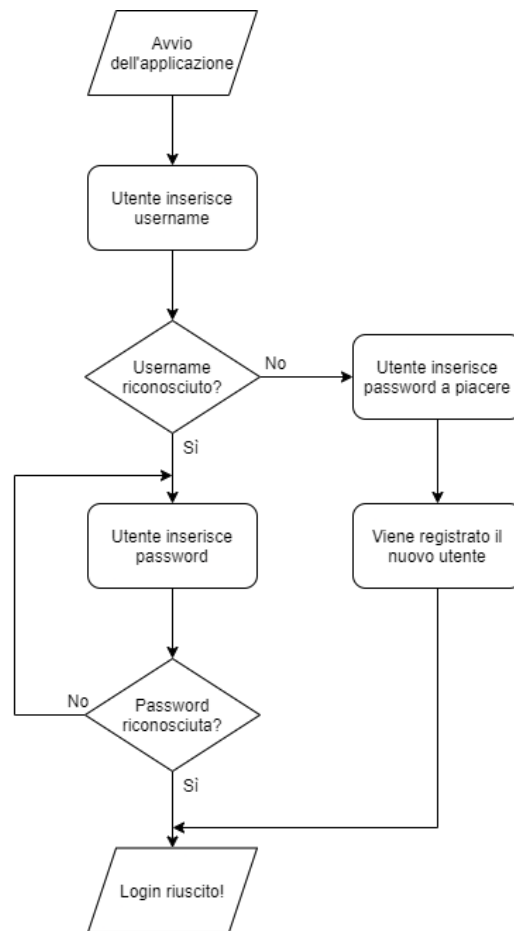


Figura 2: Login workflow

I dati vengono scritti all'interno di un dizionario che viene poi salvato su file in formato json. Se inizialmente il file non esiste, viene creato automaticamente.

4.3 Web Server

Una volta effettuato con successo il login, il Web Server diventa attivo. Esso rimane in ascolto finchè l'utente non seleziona la console e digita la combinazione `Ctrl-C`. La caratteristica principale di questo server è il *multithreading*, ovvero offre la possibilità a diversi client di connettersi contemporaneamente ad esso. Per avere informazioni dettagliate sul funzionamento, vedere i commenti nel file `"http_multithread_server.py"`.

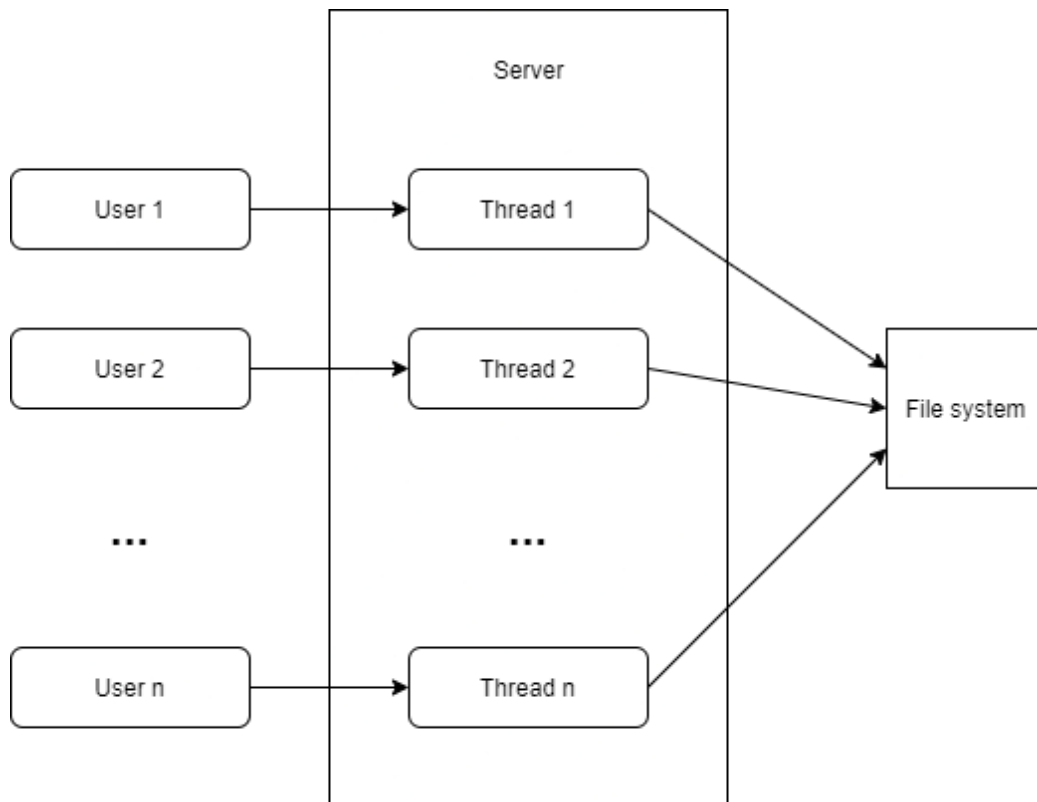


Figura 3: Struttura Web Server Multithread

In figura viene mostrata la struttura "ideale" per un web server multithread, ovvero con un file system (oppure un database) in comune con tutti gli utenti, salvato su cloud. Essendo un progetto accademico, per semplicità non c'è un database condiviso, ma ogni utente salva in locale i dati generati dall'applicazione.