

# Elaborato di Telematica

## 1 Premessa

Il corso di Telematica prevede la consegna di un elaborato da svolgersi extra-orario di lezione. L'elaborato può essere svolto in forma individuale ovvero in gruppo (max 3 persone), tutti studenti del corso di Telematica.

Per ogni gruppo deve essere indicato un referente, che tiene i contatti (principalmente via e-mail) e provvede alla consegna dell'elaborato. La valutazione assegnata all'elaborato viene attribuita a tutti i componenti del gruppo.

## 2 Modalità operative

### 2.1 Proposta di elaborato

Il referente deve inviare un'e-mail contenente una proposta di elaborato, contenente le seguenti informazioni:

- titolo del progetto;
- partecipante/i (nome, cognome, matricola);
- breve descrizione dell'idea progettuale;
- strumenti previsti per lo svolgimento (PC, sistema operativo, strumenti di sviluppo, ecc.).

Se la proposta è accettata, l'elaborato sarà ritenuto valido ed il suo sviluppo potrà procedere. Nel caso in cui non si riceva una risposta entro una settimana di tempo dall'invio della richiesta, la proposta è da considerarsi approvata.

### 2.2 Svolgimento dell'elaborato

Lo svolgimento viene effettuato dal gruppo di lavoro in forma completamente autonoma, sia per la tempistica sia per quanto riguarda la distribuzione dei compiti e la sequenza delle attività.

Tuttavia, è consigliabile sincronizzarsi con il docente a metà lavoro, in modo da verificare il corretto allineamento con gli obiettivi iniziali ed apportare eventuali cambiamenti di indirizzo nel caso in cui fossero necessari. Nella stessa ottica, è possibile scambiare opinioni e chiedere eventuali suggerimenti su punti specifici.

## **2.3 Consegna dell'elaborato**

La consegna finale dell'elaborato progettuale consiste in una relazione finale (breve) ed in una presentazione (in formato PPT) del lavoro svolto. Nel caso in cui sia previsto lo sviluppo di software, è richiesta la consegna di un archivio in formato ZIP contenente il codice sorgente ed un file che descriva come eseguire il codice consegnato.

La relazione finale deve essere in formato PDF ed inviata tramite e-mail almeno 2 settimane prima della data di consegna in cui si intende procedere alla presentazione.

La data di consegna è libera. Verranno successivamente comunicate le date possibili per la presentazione orale, in modo da raccogliere un certo numero di adesioni e creare una sessione interattiva e di interesse per tutti i partecipanti. Ogni studente avrà a disposizione 7 minuti per la presentazione, di cui 5 minuti per la presentazione del progetto e 2 minuti per eventuali domande. Per progetti svolti in gruppo, il tempo a disposizione sarà leggermente maggiore, in modo da dare la possibilità a tutti i componenti di esporre la propria parte (dovrà trasparire l'apporto individuale di ognuno al progetto).

## **3 Argomenti possibili**

In linea di principio qualsiasi proposta può essere accettata purchè sia pertinente agli argomenti del corso (in particolare deve riguardare l'aspetto dei protocolli di comunicazione). Oltre che sviluppare un progetto a sè stante, è anche possibile basarsi su lavori già iniziati o previsti nell'ambito di altri corsi di insegnamento o comunque di altre attività. In tal caso, come per le proposte elencate di seguito, deve essere chiaramente indicato l'ambito originario e la parte che si intende sviluppare per il corso di Telematica. Quindi, nel caso in cui ci fossero argomenti di interesse non compresi tra quelli riportati di seguito, sentitevi liberi di fare una proposta tramite e-mail e di discuterne insieme.

Di seguito si elencano alcune proposte, suscettibili di eventuali varianti e/o approfondimenti.

### **3.1 Elaborati che prevedono sviluppo di software**

#### **3.1.1 Validatore di URL**

Dato un insieme di pagine HTML disponibili offline ovvero online, il componente ricerca tutti i link presenti in tali pagine e verifica che siano validi.

In particolare, di ogni link dovrebbe indicare lo stato, distinguendo fra varie possibili anomalie, ad esempio:

- OK (valido, codice di risposta 200);
- OK (ma con redirectione);
- la risorsa non esiste più;
- il server non risponde (host non raggiungibile);
- connessione rifiutata (server HTTP non attivo);
- pagina privata (occorre autenticarsi).

### 3.1.2 Diagnostica HTTP

Strumento per generare richieste personalizzate in modo da esaminare le risposte di un certo server HTTP. I messaggi HTTP devono potere essere costruiti in modo preciso, agendo su ogni possibile header field.

### 3.1.3 Client POP3

Client non standard (a scelta se POP3 e/o IMAP4) che opera connettendosi ad un provider di posta elettronica, e consente l'esecuzione di differenti operazioni (es. download delle intestazioni, visualizzazione dei messaggi, cancellazione, analisi del percorso del messaggio di posta elettronica mediante gli header

field **Received**, ecc.). Altre funzionalità che possono essere istanziate dal client possono essere le seguenti:

- interrogazione di molteplici mailbox e segnalazione dell'arrivo di nuovi messaggi in ognuna di esse (le coppie username/password per ogni mailbox potrebbero essere registrate preventivamente, ad es. in un file ovvero in un database);
- segnalazione dell'arrivo di messaggi da determinati mittenti;
- estrazione automatica di allegati: l'applicativo potrebbe funzionare in background ed accedere alla/alte mailbox ad intervalli periodici. In ogni caso, dovrebbe essere in grado di:
  - distinguere fra messaggi già arrivati e messaggi nuovi;
  - decodificare correttamente i campi (quali **Subject**, **From**, **To**, ecc.) che possono contenere caratteri estesi (non ASCII).

### **3.1.4 Gateway HTTP-POP3**

Applicativo che svolge le funzionalità del client descritto al punto 3.1.3, ma a cui si accede tramite un browser Web, e che emette richieste per la consultazione di mailbox verso un server di posta esterno. Tipicamente realizzato mediante pagine in linguaggio PHP richiede la presenza di un server HTTP di appoggio.

### **3.1.5 Monitor Web**

Data una serie di pagine Web statiche segnala quando vengono modificate. Tipicamente funziona in background scandendo un elenco di URL contenuto in un file, quindi ogni URL viene analizzato con una certa frequenza e periodicamente (ovvero in tempo reale) genera un elenco (report) delle pagine che risultano aggiornate.

### **3.1.6 Web Robot**

Corrisponde allo sviluppo di un robot con un task da svolgere ben specifico (es. analizzare gli header serviti dagli host per le varie pagine ed estrarne delle statistiche; ricercare solamente documenti in una determinata lingua; ricercare solo immagini; ecc.). A scelta dello sviluppatore se fornire il robot di un'interfaccia di gestione Web-based, ovvero di un'interfaccia da command line.

## **3.2 Elaborati senza sviluppo di software**

Rientrano in questa categoria lavori che, ad esempio, prevedono la configurazione di applicativi, oppure l'analisi di prestazioni o simili.

### **3.2.1 Web Proxy**

L'idea è di attivare funzionalità di proxy su un server HTTP (es. Apache, ma è possibile utilizzare altri software). Tali funzioni dovrebbero comprendere, ad esempio:

- autenticazione dell'utente;
- abilitazioni in funzione della data/ora;
- Proxy Auto Configuration (PAC).

Un ulteriore sviluppo potrebbe essere l'attivazione di un "reverse proxy" (vedere documentazione Apache) come front-end di uno o più server HTTP.

### 3.2.2 Server HTTP

Configurazione di un server con almeno le seguenti funzionalità:

- (name-based) virtual host;
- gestione TLS;
- pagine dinamiche (es. tramite PHP).

In aggiunta, sarebbe opportuno interagire con un database SQL-based in modo da gestire un piccolo (o grande) database accessibile da Web.

### 3.2.3 WebDav

*Distributed Authoring and Versioning* (DAV) è un'estensione del protocollo HTTP avente lo scopo di consentire ad utenti autorizzati di caricare sul server le proprie pagine Web. Un primo passo in questa direzione sarebbe l'utilizzo del protocollo DAV per l'abilitazione del metodo PUT (che in genere non è implementato sui server canonici), quindi di tutte le funzionalità del protocollo DAV. Un secondo aspetto di questo progetto riguarda la parte client. Si possono utilizzare alcuni software semi-sperimentali reperibili in rete, ovvero sviluppare una soluzione *custom*.