

Relazione Tecnologie Web 2014

Andrea Giacomo Baldan, Alberto De Agostini, Matteo Rigon

18 Giugno 2014

- E-mail referenti
 - *E-mail: a.g.baldan@gmail.com*
 - *E-mail: miniotta@hotmail.it*
 - *E-mail: matteorigon@alice.it*
- Specifiche
 - *URL sito: <http://tecnologie-web.studenti.math.unipd.it/tecweb/> utente*
 - *Nome utente:*
 - *Password:*

1 Perl

Trattandosi di un sito con una buona quantità di contenuti dinamici, è stato studiato un approccio quanto più modularizzato possibile, in modo da garantire maggior chiarezza e manutenibilità, una sorta di MVC, dove le view sono rappresentate da template (`.tmpl`) raccolti in una directory completamente separata dal codice, modelli e controller sono contenuti in 3 file contenenti le funzionalità principali, si è quindi resa necessaria la suddivisione di esse in una gerarchia formata da tre moduli:

- **UTILS** classe padre, raccoglie le funzioni di uso generale per il funzionamento e la popolazione delle varie pagine, caricamento ed interfaccia dei vari database XML (Model)
- **UTILS::Admin** classe figlio di UTILS, raccoglie le funzioni strettamente necessarie al backend dell'applicazione, funzionalità di login e mantenimento delle sessioni
- **UTILS::UserService** classe figlio di UTILS, raccoglie le funzioni necessarie al compimento delle operazioni strettamente legate all'utente (e.g CRUD delle proprie generalità), prenotazione risorse

In particolare ognuno di questi moduli fa da appoggio a rispettivi script utilizzati per effettuare le varie operazione per mezzo di dispatch tables, che consentono di risparmiare un gran numero di operazioni ridondanti e di automatizzare il più possibile le operazioni da eseguire, aumentando inoltre la separazione tra codice e contenuto, avvicinandosi ad un approccio MVC:

- **load.cgi** si appoggia ad **UTILS** ed è il motore di popolamento principale del sito, ogni pagina accessibile è generata e popolata da questo script, per mezzo di dispatch tables
- **admin.cgi** si appoggia ad **UTILS::Admin**, controparte backend di **load.cgi**, ogni pagina della parte amministrativa è generata da questo script
- **process.pl** script necessario alle basilari operazioni di modifica/popolamento risorse/pagine (CRUD)
- **user_jobs.pl** controparte frontend di **process.pl**, tutte le operazioni che l'utente può effettuare sono gestite da questo codice