

MenùT_EX: una piattaforma per realizzare menù

Claudio Fiandrino

Sommario

L'articolo presenta MenùT_EX, una piattaforma che realizza semplici menù stampati su foglio A4 in formato 2×3 . Per la redazione del documento, la piattaforma mette a disposizione un'interfaccia grafica attraverso cui gli utenti accedono ad informazioni presenti su un database.

Abstract

The article presents MenùT_EX, an architecture which aims to print simple menus on A4 paper in a 2×3 format. To create the final document, the architecture provides a graphical interface that enables the users to access information stored in a database.

1 Introduzione

Un menù è un documento più semplice rispetto ad una tesi o a un articolo o a una semplice lettera. Recentemente, mi è stato chiesto se ci fosse un metodo migliore rispetto all'utilizzo dei comuni software commerciali per realizzare semplici menù da stampare in formato 2×3 su foglio a4 e la scelta è caduta subito su L^AT_EX. Perché? In primo luogo garantisce una qualità decisamente superiore rispetto ad altri software. Inoltre, il sorgente di un documento è codice, perciò può essere preparato con una *programmazione ad hoc*. Da qui è nato MenùT_EX.

Il concetto di *programmazione ad hoc* è sicuramente il pilastro fondamentale della piattaforma. Rispetto ad altri tipi di documenti, un menù è sostanzialmente composto da un testo semplice, poche righe ognuna delle quali contiene generalmente poche parole. Queste righe possono cambiare giorno per giorno, tuttavia le combinazioni non sono infinite e il loro contenuto può ripetersi nel tempo. Un'organizzazione efficiente di questo tipo di informazione richiede un database. È quindi importante che le informazioni presenti sul database siano utilizzabili di volta in volta nel documento L^AT_EX. Per questo motivo la piattaforma MenùT_EX è composta da un'interfaccia grafica che permette agli utenti di organizzare e selezionare le informazioni su un database e redigere il documento da stampare con L^AT_EX sulla base delle scelte effettuate.

I menù presi in considerazione sono molto semplici, con una doppia scelta di primi e secondi piatti più eventuali variazioni. Sono adatti, ad esempio, a ritiri di gruppi sportivi. Tuttavia, non si pensi che la piattaforma funzioni proprio per la relati-

va semplicità del progetto. L'idea base può essere tranquillamente estesa in altri contesti, ad esempio la realizzazione sistematica di badge per eventi come conferenze.

L'articolo è così strutturato. La sezione 2 illustra l'ambiente di lavoro usato per lo sviluppo e necessario per l'utilizzo della piattaforma. La sezione 3 analizza ad alto livello l'interfaccia grafica mentre la sezione 4 prende in esame il nucleo di MenùT_EX, ossia la generazione e la compilazione dei documenti L^AT_EX. Infine, la sezione 5 conclude il lavoro.

2 L'ambiente di lavoro

Il sistema operativo utilizzato per lo sviluppo è Mac OS X. Il funzionamento del sito per l'interfaccia grafica avviene in locale grazie all'applicazione open source MAMP. L'acronimo del nome è generato dalle seguenti iniziali: Mac OS X (il sistema operativo), Apache (la piattaforma server Web più comune e supportata dalla maggioranza dei sistemi operativi), MySQL (la piattaforma per la gestione dei database) e PHP (il linguaggio di programmazione "server-side" più diffuso per Web secondo <http://w3techs.com/>). I file web dell'interfaccia grafica sono stati posizionati in `/Applications/MAMP/htdocs/`.

Lo sviluppo dell'interfaccia grafica di MenùT_EX in forma Web è stata pensata con i seguenti obiettivi. Primo in ordine di importanza, non si vincolano gli utenti a questo piuttosto che quel sistema operativo. MAMP è pensato per uno specifico sistema operativo, ma esistono simili applicazioni per Linux e Windows (LAMP e WAMP rispettivamente). Tutte queste applicazioni supportano intrinsecamente piattaforme di gestione dei database e la portabilità richiede variazioni al software minime. Non è stato necessario sviluppare un apposito editor perché i parametri da selezionare possono essere inseriti tramite qualsiasi browser installato sulla propria macchina. Inoltre, la piattaforma funziona in locale, ma può essere "convertita" a sito Web remoto vero e proprio con pochissime modifiche.

Per ottenere il risultato finale con L^AT_EX si è usata la distribuzione MacT_EX, pensata appositamente per il sistema operativo Mac OS X; in particolare, per il funzionamento di MenùT_EX sono strettamente necessari la classe standalone SCHARRER (2012) e i pacchetti:

- TikZ TANTAU (2013);



(a) La pagina di input.

(b) La pagina di conferma e stampa.

FIGURA 1: L'interfaccia grafica.

- pdfpages MATTHIAS (2013);
- background MEDINA (2014).

3 L'interfaccia grafica

L'interfaccia grafica è lo strumento che permette agli utenti di gestire e selezionare le informazioni presenti sul database per la stampa.

Sviluppata con codice HTML e PHP, l'interfaccia si compone di quattro pagine principali:

- la home page;
- la pagina per selezionare le informazioni dal database;
- la pagina per immettere o eliminare informazioni dal database;
- la pagina per visualizzare le informazioni presenti nel database.

Ai fini dell'articolo prenderemo in analisi soltanto la seconda. Tale pagina non solo permette di selezionare le informazioni dal database, ma attiva anche la procedura di stampa.

La figura 1a riporta l'aspetto della pagina di selezione delle informazioni. L'utente può selezionare con classici menù a tendina le opzioni per i primi e secondi piatti. Queste informazioni vengono reperite dal database. Inoltre, è anche necessario scegliere quale tipo di menù stampare; in questo caso le opzioni sono soltanto due: *Pranzo* o *Cena*. Infine, l'utente deve digitare nel riquadro testuale il numero di menù da stampare.

Il bottone "Reset" permette di annullare tutte le opzioni selezionate. Se invece l'utente ha inserito tutti i parametri, può continuare attraverso il bottone "Submit". Una volta compiuta questa operazione, dietro le quinte viene effettuato un

controllo per verificare la presenza e la correttezza di tutti i parametri inseriti. Se il controllo ha esito positivo, si viene indirizzati verso la pagina di conferma e in automatico parte la procedura di stampa. La pagina di conferma è mostrata in figura 1b.

4 Il backend: usiamo L^AT_EX

Come si è visto, l'interfaccia grafica permette all'utente di selezionare una serie di parametri che dovranno far parte di un documento L^AT_EX. Si analizza ora la metodologia con cui MenùT_EX gestisce questi parametri. È importante notare che l'utente finale non si accorge delle operazioni che si andranno a prendere in esame. Ogni passaggio, dalla redazione alla stampa finale, viene svolto rigorosamente in background.

4.1 La preparazione

Avendo un vincolo preciso sul numero di menù da stampare per foglio (6) e sapendo che il numero complessivo di menù (n) può potenzialmente cambiare ogni giorno, si è pensato di operare dividendo il flusso di lavoro in due passi. Nel primo passo, si crea un documento di n pagine in cui ogni pagina è un singolo menù. A tale proposito, la classe *standalone* è senza dubbio lo strumento migliore. Nel secondo passo, utilizzando il pacchetto pdfpages si può facilmente raggruppare nel formato 2×3 le pagine del primo documento.

Redigere il documento standalone

La figura 2 mostra una pagina del documento standalone con i parametri di ingresso riportati in figura 1b. La "pagina-base" è stata realizzata con TikZ caratterizzando ogni etichetta come nodo e posizionando i vari nodi con l'aiuto della libreria positioning.

Menù del giorno	
7 Agosto	
Pranzo	
Gnocchi al Castelmagno	N...
Ravioli ai funghi	N...
Fritto misto	N...
Arista di maiale al forno	N...
Altro	

FIGURA 2: Una pagina del documento standalone.

La scelta del pacchetto non è casuale. In primo luogo, nonostante l'aspetto attuale del menù sia decisamente minimalistico, la sua implementazione con TikZ può permettere l'aggiunta di ulteriori elementi grafici in maniera semplice. In secondo luogo, la classe `standalone` ha un'opzione `tikz` per scontornare in modo ottimale figure composte con TikZ. Infine, TikZ fornisce un costrutto `\foreach` perciò si possono creare n pagine uguali utilizzando come idea base il seguente pseudo-codice:

```
\documentclass[tikz]{standalone}

\begin{document}
\foreach \pagina in {1,...,<n>}{
  \begin{tikzpicture}
    codice figura
  \end{tikzpicture}
}
\end{document}
```

Si noti che l'opzione `tikz` di `standalone` carica automaticamente TikZ.

Prima di redigere il documento, per prima cosa è necessario creare un nuovo file. Tale file viene posizionato in una cartella temporanea chiamata `file-menu` nel solito percorso `/Applications/MAMP/htdocs/`. Il comando PHP utilizzato è:

```
$fh = fopen("./file-menu/$nf.tex", "w")
or die("can't open file");
```

dove `$nf` è una variabile che contiene il nome del file senza estensione, `$fh` è il nome del file handler che si utilizzerà successivamente e l'opzione `"w"` specifica che il file viene aperto in scrittura. Tuttavia, per essere completamente certi di poter scrivere sul nuovo file, è buona regola specificarne i permessi:

```
chmod("./file-menu/$nf.tex",777);
```

PHP mette anche a disposizione una funzione `tempnam()` per creare file temporanei con nomi

univoci. Tuttavia, volendo creare una copia di archivio per ogni menù stampato, è preferibile procedere assegnando a `$nf` un nome significativo oltre che univoco. Menù $\TeX$ usa la data nel formato giorno-mese-anno più una stringa per distinguere se il menù sia "Pranzo" oppure "Cena".

A questo punto si può redigere il documento. Considerato il limitato numero di righe di codice, si è proceduto riga per riga salvando in una variabile temporanea la stringa e scrivendola sul file con l'aiuto del file handler. Ad esempio:

```
$strData = "\usepackage[T1]{fontenc}\n";
fwrite($fh, $strData);
```

Un'attenzione particolare deve essere prestata agli elementi che l'utente deve selezionare come parametri. Ad esempio, il codice per descrivere la prima opzione del primo piatto è:

```
$strData = "\\node[text width=7.2cm,
  align=left,
  anchor=south west]
  (scelta-1) at (0.25,4.5)
  {{{$primo_piatto_op1}}};\n";
fwrite($fh, $strData);
```

Occorre prestare attenzione al fatto che in PHP il carattere `\` deve essere "protetto" in maniera simile a quanto avviene con L $\TeX$. Inoltre, è anche necessario raddoppiare le parentesi graffe: analogamente con il caso precedente, quelle interne "proteggono" il contenuto. Senza tale operazione, PHP scriverebbe sul file soltanto il valore della variabile `$primo_piatto_op1` senza le parentesi graffe (generando di conseguenza un errore). Questa variabile, come ogni parametro di input, rappresenta il valore selezionato dall'utente e viene trasferito dalla pagina di inserimento dei parametri a quella di stampa e conferma attraverso il metodo POST.

Il codice 1 mostra il sorgente del documento standalone completo necessario a generare le pagine illustrate nella figura 2 di esempio.

Redigere il documento finale

Completato il documento standalone, si può procedere alla redazione del documento finale dopo avere creato il nuovo file e specificato i permessi con operazioni del tutto analoghe a quanto descritto in precedenza. La figura 3 mostra una pagina di esempio.

Come accennato, per ottenere la disposizione 2×3 è conveniente caricare il pacchetto `pdfpages`. Il comando base `\includepdf` permette di assegnare all'opzione `nup` il valore `2x3`. Quindi:

```
$strDataN = "\includepdf[pages=-,
  nup=2x3,
  offset=-0cm 0cm,
  delta=0.4cm 0cm]
  {{{$nf.pdf}}} \n";
fwrite($fh_n, $strDataN);
```

```

\documentclass[tikz,border=2pt]{standalone}
\usepackage[T1]{fontenc}
\usepackage[utf8]{inputenc}
\usepackage[italian]{babel}
\usepackage{charter}
\usepackage{tikz}
\usetikzlibrary{positioning}

\tikzset{posizione/.style={align=left,anchor=south west}}

\begin{document}
\foreach \num in {1,...,25}{
\begin{tikzpicture}[font=\LARGE]
\useasboundingbox (0,0) rectangle (9,8);
\path (0,0) rectangle (9,8);

% INTESTAZIONE
\node (menu) at (4.5,7.25){Men\'{u} del giorno};
\node[below = 0cm of menu](date) { 7 Agosto};
\node[below = 0.075cm of date,font=\Large] (pranzo) {Pranzo};

% PIATTI
\begin{scope}[font=\Large,minimum height=1ex,text height=2ex]
% PRIMI
\node[text width=7.2cm,posizione] (scelta-1) at (0.25,4.5){Gnocchi al Castelmagno};
\node[text width=1cm,posizione,right= 0cm of scelta-1] (numero-1) {N\,...};
\node[text width=7.2cm,align=left,below= 0cm of scelta-1] (scelta-2)
{Ravioli ai funghi};
\node[text width=1cm,posizione,right= 0cm of scelta-2] (numero-2) {N\,...};
% SECONDI
\node[text width=7.2cm,posizione,below=0.2cm of scelta-2] (scelta-3) {Fritto misto};
\node[text width=1cm,posizione,right= 0cm of scelta-3] (numero-3) {N\,...};
\node[text width=7.2cm,align=left,below= 0cm of scelta-3] (scelta-4)
{Arista di maiale al forno};
\node[text width=1cm,posizione,right= 0cm of scelta-4] (numero-4) {N\,...};
% ALTRO
\node[text width=7.2cm,posizione,below=0.3cm of scelta-4] (altro)
{Altro .....};
\end{scope}

\end{tikzpicture}
}
\end{document}

```

CODICE 1: Il documento standalone

Menù del giorno 7 Agosto Pranzo Gnocchi al Castelmagno N ... Ravioli ai funghi N ... Fritto misto N ... Arista di maiale al forno N ... Altro	Menù del giorno 7 Agosto Pranzo Gnocchi al Castelmagno N ... Ravioli ai funghi N ... Fritto misto N ... Arista di maiale al forno N ... Altro
Menù del giorno 7 Agosto Pranzo Gnocchi al Castelmagno N ... Ravioli ai funghi N ... Fritto misto N ... Arista di maiale al forno N ... Altro	Menù del giorno 7 Agosto Pranzo Gnocchi al Castelmagno N ... Ravioli ai funghi N ... Fritto misto N ... Arista di maiale al forno N ... Altro
Menù del giorno 7 Agosto Pranzo Gnocchi al Castelmagno N ... Ravioli ai funghi N ... Fritto misto N ... Arista di maiale al forno N ... Altro	Menù del giorno 7 Agosto Pranzo Gnocchi al Castelmagno N ... Ravioli ai funghi N ... Fritto misto N ... Arista di maiale al forno N ... Altro

FIGURA 3: Una pagina del documento completo.

```

\documentclass{article}
\usepackage[T1]{fontenc}
\usepackage[utf8]{inputenc}
\usepackage[italian]{babel}
\usepackage{charter}

\usepackage{pdfpages}
\usepackage{tikz}
\usetikzlibrary{calc}
\usepackage[contents={}] {background}

\newcommand{\pageindicators}{%
\tikz[remember picture,overlay]{%
\draw[ultra thin, dash pattern=on 9.5pt off 9pt]
(current page.north)--(current page.south)
($ (current page.north west)!0.6675!(current page.west)$)--
($ (current page.north east)!0.6675!(current page.east)$)
($ (current page.south west)!0.6675!(current page.west)$)--
($ (current page.south east)!0.6675!(current page.east)$);
}%
}

\AddEverypageHook{%
\pageindicators%
\BgMaterial}%

\begin{document}
\includepdf[pages=-,nup=2x3, offset=-0cm 0cm, delta=0.4cm 0cm]
{Pranzo-7-08-2014-single.pdf}
\end{document}

```

CODICE 2: Il documento finale.

Ovviamente, è necessario caricare il file pdf del sorgente `.tex` creato in precedenza. Non specificando l'estensione del file direttamente all'interno della variabile `$nf`, quest'ultima può essere riutilizzata senza problemi.

Per agevolare il taglio dei vari menù di ogni singola pagina, si è pensato di introdurre dei segmenti di delimitazione. Ancora una volta, TikZ è di aiuto con un nodo speciale, `current page`, il quale identifica completamente la pagina che si compone. Ad esempio, per accedere al punto in alto a sinistra del foglio, è sufficiente specificare `current page.north west`.

Per poter *aggiungere* elementi alla pagina corrente generalmente si ricorre alle opzioni `remember picture` e `overlay`. Tali opzioni hanno un inconveniente: richiedono una doppia compilazione. In seguito si ritornerà su questo punto. Il nodo `current page` inoltre, identifica di volta in volta la pagina corrente. Tuttavia, i segmenti devono essere presenti in tutto il documento perciò si è usato il pacchetto `background`, sviluppato appositamente per questo genere di necessità. In genere si procede in questo modo. Per prima cosa si raggruppano in una macro gli elementi da inserire in ogni pagina (`\pageindicators`). Successivamente si utilizza la macro appena definita come parametro:

```

\AddEverypageHook{%
\pageindicators%
\BgMaterial}%

```

Il codice 2 mostra il sorgente del documento

finale completo necessario a generare le pagina di esempio riportata in figura 3.

4.2 La compilazione

Per ottenere i documenti mostrati nelle figure 2 e 3 è ovviamente necessario compilare i sorgenti. Il primo requisito in questa fase è l'obbligo di dover compilare due volte il sorgente del documento finale in quanto si sono usate le opzioni `remember picture` e `overlay` in una figura TikZ. Inoltre, si deve prestare attenzione affinché il documento standalone sia compilato *prima* del documento finale.

PHP permette l'esecuzione di programmi esterni con diverse funzioni. Nel caso specifico, poiché il motore di composizione `pdflatex` può essere eseguito da terminale, si è scelto di usare la funzione `shell_exec()`.

Per semplicità, si è scelto di creare un piccolo script bash capace non solo di compilare i sorgenti nell'ordine esatto, ma anche di rimuovere i file ausiliari e i sorgenti dopo il processo di compilazione. Il codice dello script è il seguente:

```

#!/bin/bash

cd ./file-menu
# doppia compilazione
for file in *.tex; do
    /usr/local/texlive/2014/bin/
    x86_64-darwin/pdflatex "$file";
done
for file in *.tex; do

```

```
/usr/local/texlive/2014/bin/
x86_64-darwin/pdflatex "$file";
done
# rimozione file ausiliari e sorgenti
rm *.log
rm *.aux
rm *.tex
```

Come primo passo si accede alla cartella in cui i sorgenti sono posizionati e, per ogni file con estensione `.tex`, si lancia il processo di compilazione. Questa scelta si rivela funzionale perché i sorgenti vengono rimossi una volta terminato il processo. In questo modo si è sicuri che per ogni menù che si genera, la cartella `file-menu` inizialmente *non* contiene nessun sorgente. Il rispetto del vincolo di precedenza nella compilazione, invece, viene garantito dietro le quinte attraverso la scelta dei nomi dei due sorgenti.

Si noti che è stato necessario specificare *per intero* il percorso del motore di composizione, pena il non funzionamento dello script. Per non specificare completamente l'intero percorso e quindi avere una versione semplificata dello script:

```
#!/bin/bash

cd ./file-menu
# doppia compilazione
for file in *.tex; do
  pdflatex "$file";
done
for file in *.tex; do
  pdflatex "$file";
done
# rimozione file ausiliari e sorgenti
rm *.log
rm *.aux
rm *.tex
```

sarebbe necessario configurare il file `.bashrc` in modo opportuno affinché i percorsi dei motori di composizione siano riconosciuti.

5 Conclusione

L'articolo ha illustrato MenùT_EX, una piattaforma che realizza semplici menù stampati su foglio a4 in

formato 2×3. Si è focalizzata l'attenzione sull'architettura, evidenziando l'interazione tra l'interfaccia grafica e il backend, ossia come avviene la creazione dinamica dei sorgenti L^AT_EX con parametri impostati dall'utente e la loro successiva compilazione. Il backend è completamente nascosto all'utente che quindi non deve necessariamente avere conoscenze L^AT_EX specifiche per poter operare.

MenùT_EX è oggi praticamente utilizzato dal ristorante Tre Verghe d'Oro (<http://www.treverghe.it/>), perciò si può considerare a tutti gli effetti un esempio concreto di applicazione L^AT_EX in un contesto commerciale.

6 Ringraziamenti

Voglio ringraziare il revisore anonimo per i preziosi suggerimenti che contribuiranno sicuramente al miglioramento della piattaforma. Un doveroso ringraziamento va a Claudio Beccari, il direttore di *ArSTeXnica* e a tutta la redazione per il prezioso lavoro e la passione che dedicano alla rivista.

Riferimenti bibliografici

- MATTHIAS, A. (2013). *The pdfpages Package*. URL <http://www.ctan.org/pkg/pdfpages>. Consultabile con: `texdoc pdfpages`
- MEDINA, G. (2014). *The background package*. URL <http://www.ctan.org/pkg/background>. Consultabile con: `texdoc background`
- SCHARRER, M. (2012). *The standalone Package*. URL <http://www.ctan.org/pkg/standalone>. Consultabile con: `texdoc standalone`
- TANTAU, T. (2013). *The TikZ and PGF Packages*. URL <http://www.ctan.org/pkg/pgf>. Consultabile con: `texdoc tikz`

▷ Claudio Fiandrino
claudio dot fiandrino at gmail
dot com