

Una guida alla preparazione di documenti estesi

elaborato, tesi(na), libro,...

M.Bertocco

ver.2.1 – 12 novembre 2009

Copyright © Matteo Bertocco, 2009

La presente guida può essere liberamente riprodotta e distribuita, a patto che non siano esercitati fini o diritti commerciali e che la stessa venga riprodotta nella sua forma integrale senza alcuna modifica e includendo la presente nota di copyright.

La disattesa alle condizioni precedenti viene considerata violazione al diritto di copyright esercitato dall'autore.

Prefazione

Caro lettore,
vai di fretta?

Allora ti prego di saltare direttamente al capitolo 5. E poi prova praticamente a scrivere qualcosa con L^AT_EX.

Se durante la scrittura ti vengono in mente delle domande sul “come fare per...” (es. “come includo una figura?”), allora puoi consultare l’indice analitico in fondo per trovare dei suggerimenti.

Ancora qui? Bene allora, se non sei troppo di fretta il capitolo 2 ti attende...

Se stai leggendo anche questa riga, allora ringrazio per la pazienza, puoi proseguire con la lettura in buon ordine, nella speranza che questa guida possa essere di aiuto. Suggerimenti al suo miglioramento sono i benvenuti.

L’autore.

p.s.

La prefazione è un commento al testo e/o anche all’autore, e viene normalmente scritta da una terza persona. Tale persona è tipicamente competente in materia, e quindi, salvo prova contraria, *non* si deve chiedere di scrivere la prefazione alla nonna o agli amici della squadra di calcetto, (nell’ultimo caso a meno che non si tratti del libro “soccer for dummies”!)

Contenuti possibili di una prefazione includono le caratteristiche attese del lettore (*a chi* è rivolto il lavoro, e quali competenze dovrebbe avere per capirlo), il “taglio” dato al lavoro (testo scientifico, testo a carattere didattico, romanzo, ...), un breve cenno agli “obiettivi di apprendimento” che il lavoro dovrebbe permettere di conseguire (*che cosa* imparerà il lettore),

o ancora l'uso atteso del testo (*come* va letto, ossia in che ordine, *quando* svolgere esercitazioni e come, ecc.)

La prefazione potrebbe anche contenere elenco sintetico dei contenuti del testo, e tuttavia tale elenco dovrebbe limitarsi all'esposizione dei temi trattati (es. se il testo è diviso in parti, allora vengono citate le parti con una introduzione ai contenuti di ciascuna non superiore a un paio di righe). Per un eventuale commento dettagliato alle parti e ai contenuti dei capitoli è più conveniente sfruttare l'introduzione, la quale invece è bene inquadrare il tema trattato (la materia del contendere), il contesto rispetto al quale il testo si riferisce, i contributi attesi, ecc.

Piuttosto che trasformare la prefazione in un'introduzione è meglio non scriverla affatto. E in ogni caso si raccomanda di redigerla *per ultima*.

Infine, la prefazione *va firmata*.

Sommario

(o Piano dell'opera)

Lo scopo di questa guida è di aiutare sotto il profilo “editoriale” un autore all'impiego di \LaTeX nella preparazione di un libro o qualcosa di simile.

Non è una guida su \LaTeX , per questo ci sono bei libri, documenti scritti meglio di questo ecc.

Però, capita di dover iniziare a scrivere un documento esteso, e allora sorge il dubbio di come sfruttare al meglio \LaTeX .

Queste cose non si trovano sulle guide di \LaTeX , o sui testi di riferimento. In questi ultimi sono spiegate cose come le *macro* o consimili. Ma di come suddividere in più file il lavoro (e *conviene!*) o come preparare le figure ... in genere non se ne parla proprio.

Conseguentemente, si è scelto di suddividere l'esposizione in due parti.

La *prima parte* riporta come organizzare il lavoro. Vengono inoltre illustrati alcuni stili e macro disponibili per la preparazione del lavoro ecc.

Nella *seconda parte* si mostra qualche esempio concreto di impiego di stili e comandi brevemente descritti nella prima parte.

Indice

Prefazione	III
Sommario	V
1 Introduzione	1
1.1 Per iniziare	1
1.1.1 Perché \LaTeX	1
1.1.2 Che cosa serve	3
1.1.3 Documentazione	4
1.1.4 Come procedere	4
I Regole	5
2 Impostazioni	7
Introduzione	7
2.1 Organizzazione su disco	7
3 Come fare	11
Introduzione	11
3.1 File template	11
3.2 Comandi aggiuntivi (macro)	12
3.2.1 Stili	12
3.2.2 Caratteri speciali	13
3.2.3 Figure	13
3.2.4 La famiglia di macro <code>\Fig...</code>	14
3.2.5 La macro <code>\Figure</code>	14
3.2.6 Impiego di comandi del package <code>graphicx</code>	15
3.2.7 Indice analitico	16
3.2.8 Copertina	16
3.3 Riservatezza del documento	17

3.3.1	Un esempio	18
4	Tabù	27
	Introduzione	27
4.1	Elenco dei tabu	27
4.1.1	Uso della lingua italiana	27
4.1.2	Stili (tipografici)	29
4.1.3	Figure	29
4.1.4	Tabelle	31
4.1.5	Bibliografia	31
4.1.6	Cross-reference	32
II	Esempi	35
5	Per chi ha fretta	37
	Introduzione	37
5.1	Al lavoro!	37
5.1.1	Un primo test	37
5.1.2	Modificare la copertina	38
5.1.3	Scrivere un nuovo capitolo	38
5.1.4	Per proseguire...	38
5.2	Come fare per...	38
5.2.1	Cambiare il nome “elaborato”	38
5.2.2	Non usare la linea comandi	39
5.2.3	Inserire una figura	39
6	Esempi	43
	Introduzione	43
6.1	Figure	43
6.1.1	Impiego di comandi del pacchetto <code>\graphicx</code>	43
6.1.2	macro <code>\Fig...</code>	44
6.1.3	macro <code>\Figure</code>	49
	Bibliografia	53
	Elenco delle tabelle	54
	Elenco delle figure	56
	Indice analitico	58

Capitolo 1

Introduzione

Nota preliminare importante.

Il presente documento è in “evoluzione continua”. La data riportata sulla copertina permette di verificare se la versione disponibile è aggiornata rispetto a quanto reso disponibile online.

1.1 Per iniziare

Ci sono alcuni quesiti preliminari:

- perché usare \LaTeX ?
- che cosa mi serve?
- dove trovo guide o più in generale documentazione?
- in quale ordine procedo?

1.1.1 Perché \LaTeX

Non è lo scopo di questo documento argomentare circa le buone ragioni per usare \LaTeX anziché un qualunque altro tool (Nel caso di una tesi/tesina, semplicemente potrebbe venire imposto *tout-court*, per uniformare il formato nel quale le tesi/tesine sono scritte).

L’esperienza diretta dell’autore, e di molti altri, è che \LaTeX presenta i seguenti vantaggi:

- il risultato finale ha una qualità (in stampa) eccellente;

- solleva chi scrive dal peso di dover formattare un documento pensando a stili, spaziature tra paragrafi e testo, rientri, come impaginare figure (e non si rischia di veder fluttuare penosamente figure a metà tra una pagina e il vuoto), e una miriade di altre operazioni noiose o fonte di distrazione e che altri word processor richiedono. Di conseguenza è semplicemente molto più produttivo del programma “word” di turno;
- automatizza molte operazioni semplicemente penose in altri ambienti (es. creazione di sommario, indice di tabelle, figure e indice analitico);
- l’output non dipende dalla stampante scelta o in generale da driver di stampanti o da “service pack” assortiti;
- i documenti scritti una volta non devono essere mai ribattuti a fronte di aggiornamenti del software o del sistema operativo;
- il formato con i quale vengono scritti i documenti è un file testo (e quindi, non richiede un particolare editor, è possibile sfruttare facilmente ambienti consolidati per il lavoro cooperativo o per il controllo automatico di versione, come subversion, o ancora programmi realizzati ad hoc al fine di assemblare automaticamente documenti più grandi a partire da singole porzioni mantenute simultaneamente da più persone);
- è veramente affidabile: (non è così nel caso di altri programmi, capaci di far “evaporare” cross-reference o impaginazioni penosamente costruite in ore di lavoro);
- è efficiente: riformattare un documento di 400 pagine con indice analitico, parecchie decine di figure tabelle e centinaia di formule richiede un tempo ragionevole (dell’ordine del minuto con un personal computer di classe pentium 4). E comunque il documento può essere suddiviso in tanti documenti, assemblati tutti assieme all’ultimo momento;
- produce facilmente file postscript e pdf;
- è gratuito;
- è documentato in modo veramente ricco (libri, materiale online, ecc.);
- è largamente impiegato dalla comunità scientifica;

- è portabile su molte piattaforme (windows, unix, mac);
- è estensibile tramite macro;

1.1.2 Che cosa serve

Per operare in modo efficiente, è comodo disporre di alcuni strumenti software.

Qui si farà riferimento a GNU/Linux e windows.

Nel caso di *Linux*, \LaTeX è incluso nelle principali distribuzioni (il pacchetto corrispondente si chiama usualmente `texlive` o `tetex`); basta quindi solo usarlo. È però comodo disporre di un text-editor che si interfacci facilmente a \LaTeX e che sia di aiuto nella gestione di progetti costituiti da molti file. Un buon programma open-source è Kile.

Nel caso di windows, è invece necessario installare alcuni programmi, non inclusi nel sistema, ma ancora free software con licenza GPL. Tra i vari possibili si segnalano quelli disponibili gratuitamente in rete. Si consiglia la loro installazione nell'ordine indicato qui di seguito.

1. Ghostscript (programma per la gestione di file postscript);
2. Ghostview (viewer di file postscript);
3. MikTeX (implementazione di \LaTeX per windows);
4. TeXnicCenter (text editor specificamente pensato per \LaTeX), (un'alternativa è Led, invece un buon “shareware” è WinEdt);

è inoltre consigliato (anche se non strettamente necessario) installare:

- la suite `ImageMagick`, il quale contiene il programma `convert`, utile per la conversione di file png/gif/jpeg in formato encapsulated postscript;
- la suite `OpenOffice.org`. Di tale suite il programma `draw` (o anche `impress`) consente agevolmente di generare file nel formato “encapsulated postscript”, utili per la inclusione di immagini; alternative possibili in licenza GPL sono `inkscape` (programma di disegno vettoriale), `xfig` (idem, un po' datato, ma genera direttamente codice \LaTeX), ed inoltre `gimp` (per la manipolazione di immagini di tipo fotografico).

1.1.3 Documentazione

Per iniziare, si possono leggere le guide “Impara \LaTeX ” ([1]), oppure “The (not so) short introduction to \LaTeX ” ([2]). Un buon testo di riferimento è quello di L.Lamport [3].

In rete vi sono archivi (ctan, dante.de,...) ricchi di documentazione.

Infine, una volta installato \LaTeX (texlive o miktex che sia) in una directory del tipo `.../texmf/doc/` si possono trovare numerosi documenti (incluse le guide precedenti).

1.1.4 Come procedere

Per iniziare “da zero” è conveniente procedere come segue:

- leggere i documenti indicati ([1], [2]);
- installare i programmi di cui sopra;
- provare a scrivere un semplice testo (due righe sono sufficienti!) seguendo quanto riportato in [1];
- leggere questo documento e provare ad aprire i corrispondenti file che lo producono nella sua versione pdf (cioè i “sorgenti” \LaTeX);
- seguire i consigli riportati in questa guida;
- ma soprattutto, *pensare*, prima di scrivere, e *rileggere* una volta scritto.

Parte I

Regole

Capitolo 2

Impostazioni

Introduzione

In questo capitolo vengono forniti alcuni suggerimenti relativi a come impostare il lavoro. A tal fine, si suppone il lettore abbia già acquisito familiarità con \LaTeX , ad esempio a valle della lettura attenta di guide introduttive o più in generale testi specifici. Per ulteriori riferimenti in proposito si rinvia alla bibliografia.

2.1 Organizzazione su disco

Nel caso di un documento complesso, ovvero costituito da numerose pagine, figure, tabelle è conveniente cercare di procedere con ordine. Infatti, \LaTeX impone di associare a ciascuna figura un file distinto. È inoltre spesso comodo tenere una tabella in un file distinto rispetto a un file testo “principale” e che lo include. In tal modo risulta più agevole compiere operazioni di rifinitura estetica sulla tabella senza interferire con il testo, e viceversa in fase di scrittura del testo la lettura non incespica su un insieme di comandi \LaTeX che di fatto possono far perdere la concentrazione dedicata alla stesura della parte discorsiva.

In buona sostanza, quando un documento è costituito da molte pagine, allora il numero di file che concorrono alla sua definizione può crescere, e risulta allora necessario organizzare il proprio lavoro con un minimo di ordine.

Di seguito si riportano alcuni suggerimenti (caldamente consigliati) in proposito.

- suddividere il testo in file, ciascuno contenente un singolo capitolo;
- salvare/esportare le figure quando possibile (cioè sempre) in formato “encapsulated postscript” (`.eps`);
- memorizzare i file `eps` contenenti figure in una sotto-directory (nome consigliato `images`) di quella contenente il file testo al quale ciascuna figura si riferisce;
- memorizzare i file che hanno dato origine a una data figura in formato `eps` in una directory distinta da quella contenente il file `eps` da includere (nome consigliato `source-images`);
- scrivere le tabelle in file (`.tex`) distinti, e memorizzarli ciascuno in una sub-directory di quelle contenente il file che include la ciascuna tabella (nome consigliato `tables`);
- utilizzare file `BibTeX` per la gestione di riferimenti bibliografici; memorizzare i riferimenti per gruppi “omogenei” in file distinti del tipo `bib` (ad esempio libri di informatica in un file, articoli scientifici sulla compatibilità elettromagnetica in un altro ecc.);
- memorizzare i vari file `bib` in un’unica directory (nome consigliato `bib`);
- in una directory `test` memorizzare eventuali file `LATEX`ausiliari non necessari alla redazione del testo nel suo complesso nella versione finale; ad esempio, in tale directory si potrebbe memorizzare un file dal nome `test-fig.tex` che contiene solo comandi di inclusione delle varie figure impiegate nel testo, in modo da poterle stampare tutte assieme e compiere così verifiche circa la coerenza dei formati grafici adottati;
- in una directory di nome `res` memorizzare gli altri file “di servizio” utilizzati, come ad esempio i file di definizione delle macro specificamente definite per il progetto corrente, o le definizioni del tipo `hyphenation` ecc.;
- memorizzare in una directory (ad es. `backup`) copie di recupero di tutto il lavoro, sfruttando programmi di archiviazione, come ad esempio `zip` o oppure `tar/gzip`.

- memorizzare in una directory (`template`) file del tipo “modello” da copiare, rinominare e modificare quando necessario.

Capitolo 3

Come fare

Introduzione

In questo capitolo vengono brevemente illustrati:

- organizzazione dei file \LaTeX utili per la preparazione del testo;
- alcuni comandi (macro) definiti in forma aggiuntiva rispetto a quelli “di default” di \LaTeX ;

messi a disposizione per la scrittura del testo.

I corrispondenti sorgenti sono memorizzati a nella directory `res`.

3.1 File template

Una vista semplificata delle directory caldamente consigliate è riportata nella tabella 3.1.

Le tabelle 3.2 forniscono invece una descrizione sintetica dei principali file già pronti e disponibili. Tali file in alcuni casi non vanno modificati, e invece sono semplicemente inclusi così come sono.

Si noti che il presente documento è stato preparato rispettando tale schema.

Inoltre, è anche reso disponibile un archivio compresso che contiene la struttura precedente i file necessari e alcuni di parzialmente scritti che vanno modificati aggiungendo testo utile.

3.2 Comandi aggiuntivi (macro)

Per semplificare alcune operazioni frequenti, sono stati definiti alcuni comandi aggiuntivi (indicati in questo documento anche con il termine *macro*)¹.

Tali comandi definiscono ad esempio stili (cioè modificano le caratteristiche dei font), o ancora agevolano l'inserzione di figure ecc.

Nel seguito vengono presentati i comandi disponibili suddividendoli logicamente rispetto ai compiti per i quali sono stati pensati.

3.2.1 Stili

Risulta spesso utile evidenziare variamente del testo o più in generale cambiarne lo stile. Come noto, \LaTeX dispone di parecchi comandi per alterare lo stile associato al testo, come ad esempio quello per produrre caratteri in corsivo, grassetto sottolineato ecc.

Tuttavia, è *caldamente sconsigliato* utilizzare direttamente i comandi corrispondenti. Il motivo è molto semplice. Solitamente, si ricorre a una modifica di stili per evidenziare concetti, definire nuovi termini, sottolineare parole prese a prestito da altre lingue e così via.

Siccome nel ciclo di vita dell'intero documento potrebbe risultare necessario modificare lo stile adottato a seconda del tipo di "evidenziazione" che si vuole dare a una porzione di testo, è conveniente associare a comandi che ricordino il tipo di evidenziazione che si vuole dare al testo un particolare tipo di stile. A tal fine sono già disponibili alcune macro. La tabella 3.3 riporta sinteticamente il nome della macro e il significato corrispondente.

Si consiglia pertanto di usare tali comandi (macro) per cambiare lo stile di un carattere e di *non* usare direttamente i comandi \LaTeX di alterazione dello stile di caratteri.

Infine, è anche disponibile un "environment" di nome *quoteStandard*, pensato per riportare (copiare) parti di testo tratti da norme. Si usa come un qualunque altro "environment"

```
\begin{quoteStandard}    ...    \end{quoteStandard},
```

e il testo racchiuso viene indentato e evidenziato con un carattere speciale.

¹i comandi sono definiti nella classe *gmeepd*, e però il file corrispondente *non deve essere modificato*. Eventuali comandi ulteriori possono essere invece definiti nel file `res/usercmd.input.tex`.

questo è un esempio dell'effetto assunto dal testo all'interno dell'environment `quoteStandard`. L'esempio termina qui.

3.2.2 Caratteri speciali

Sono anche disponibili alcuni “caratteri speciali”, definiti tramite delle macro. La tabella 3.4 li riassume.

Ad esempio, le virgolette rispettivamente aperte e chiuse, sono definite dalle macro `\qmo` e `\qmc`² rispettivamente. Si noti che tali macro vanno impiegate facendole seguire da una coppia di parentesi graffe aperte e chiuse (`{}`), in modo separare i caratteri stessi dal testo seguente.

Si consiglia caldamente l'impiego di tali comandi anziché inserire le virgolette stesse in modo esplicito, perché ciò garantisce uniformità nel testo del loro impiego analogamente a quanto fatto con gli stili tramite i comandi corrispondenti. Inoltre sono possibili operazioni di “ridefinizione globale” delle virgolette stesse senza doverle cercare in svariati file.

Infine, la tabella 3.4 riporta per comodità alcuni comandi “standard” di \LaTeX di uso frequente.

In particolare, si ricorda come che il comando (standard \LaTeX) `\par` è equivalente a una linea bianca, e provoca dunque l'inserzione di un “a capo”, senza l'inserzione di ulteriori spaziature verticale prima del capoverso seguente. Invece, l'impiego di una doppia barra retroversa (`\\`) seguita da una riga bianca, provoca un “a capo” in cui la riga seguente viene indentata con l'aggiunta di una spaziatura verticale.

Per maggior chiarezza si mostra un esempio in proposito.

questa riga è stata preceduta dal comando `\par`

La riga sopra a questa è stata terminata con `\\` e fatta seguire da una riga bianca. Quindi tra questa riga e la precedente c'è una spaziatura verticale.

3.2.3 Figure

Nel seguito si suppone che una figura (il file corrispondente a una figura) sia contenuta nella directory `images` o in una sua sub-directory.

²dove `qmo` e `qmc` sono acronimi di *quotation mark opening* e *quotation mark closing*, rispettivamente.

Si distinguono alcune modalità fondamentali utili per l’inserzione di figure.

3.2.4 La famiglia di macro `\Fig...`

La tabella 3.5, riporta la sintassi di macro disponibili per l’inserimento di figure. Per la sintassi di ciascuna macro, si veda invece la tabella 3.6.

Per verificare il corrispondente effetto si rinvia alla sezione relativa agli esempi.

3.2.5 La macro `\Figure`

La macro `\Figure` permette di inserire una figura in modo semplice.

La sua definizione è:

```
\Figure[optionList]{caption}
```

dove:

`caption` è il testo della didascalia, mentre `optionList` è una lista di opzioni del tipo `option=value` (eventualmente racchiudendo `value` tra parentesi graffe). Le opzioni disponibili sono riportate in tab-??

A titolo di esempio per includere la figura contenuta nel file `prova.eps` (nella directory `images`) è sufficiente scrivere:

```
\Figure[file=prova]{...}
```

dove al posto dei puntini va scritta la didascalia.

Invece, per associare a una figura l’etichetta `fig:prova`, la didascalia da inserire nell’elenco delle figure “did. breve”, e scalarla i modo che sia larga 2cm è necessario scrivere:

```
\Figure[file=prova,shortcaption={did. breve},width=2cm]{...}
```

Si noti che *l’impiego della macro `\Figure` è la scelta consigliata.*

3.2.6 Impiego di comandi del package `graphicx`

Qalora con le macro definite in precedenza non si raggiungano i risultati desiderati (ad esempio perché semplicemente non previsti dalle macro), risulta necessario ricorrere a comandi \LaTeX espliciti.

A tal fine il package `graphicx`, già caricato dalla classe `gmeepd` permette di inserire figure a partire da file. Il formato *caldamente consigliato* è `encapsulated postscript` (o al più `postscript`).

Se il file è disponibile in un altro formato, si consiglia di usare il programma `convert`, tratto dalla suite liberamente acquisibile in rete `ImageMagick`.

Si sconsiglia altrettanto caldamente (da leggersi come “vietato”) invece usare il formato `jpeg` per figure del tipo “disegno a tratto”, oppure “schema a blocchi”, infatti il formato `jpeg` è poco adatto a rappresentare immagini a bordi netti, e in fase di stampa produce risultati con qualità scadente, a meno di non ricorrere a risoluzioni molto elevate (ad es. 600 dpi o più).

La documentazione a corredo del package `graphicx` è il riferimento. Alcuni esempi di impiego sono inoltre riportati nel capitolo dedicato agli esempi, in particolare al paragrafo 6.1. In questa sede si ricorda solamente che tale pacchetto fornisce tra gli altri il comando `includegraphics`, il quale permette di inserire un file (preferibilmente `eps`). Un semplice esempio di impiego di tale comando è il seguente:

```
\begin{figure}[h]
\begin{center}
\includegraphics{images/file.eps}
\caption[~short caption...]{caption...}
\label{fig:...}
\end{center}
\end{figure}
```

Il frammento \LaTeX precedente crea un oggetto fluttuante (`\begin{figure}`) il quale verrà posizionato da \LaTeX non appena possibile e compatibilmente con gli altri elementi che costituiscono il documento in modo da dare al tutto un aspetto gradevole. Ad esempio \LaTeX cercherà di evitare di posizionare la figura a metà tra due pagine, oppure in modo che sia posta subito sotto il titolo di un paragrafo. All’interno dell’oggetto fluttuante ulteriori elementi vengono centrati (`\begin{center}`). Quindi viene inserita l’immagine tramite il comando `\includegraphics{...}`, ad essa è associata una didascalia e infine una label. Si noti che il testo indicato come “short caption...” viene riportato nell’elenco delle figure, mentre

il testo riferito come “caption. . .” è la didascalia vera e propria della figura.

Qualunque sia il metodo di inserzione di figure si raccomanda di seguire le seguenti ulteriori avvertenze:

- preferire per la compilazione gli step: $\text{\LaTeX}\rightarrow\text{dvi}\rightarrow\text{ps}\rightarrow\text{pdf}$
- Nella scelta del nome da dare al file contenente le figure *non* utilizzare il carattere spazio e *underscore*, poiché possono generare errori in compilazione. Se possibile porre ancheh attenzione all’impiego di caratteri maiuscoli/minuscoli.
- Preferire solo caratteri minuscoli: ciò migliora la compatibilità tra ambienti Windows e Unix.
- Ai fini delle cross-reference si raccomanda di utilizzare label uguali al nome del file `eps` (senza estensione). Se nell’impiegare uno dei comandi del tipo `\Figure...` si è scelta la label “etichetta”, allora la stessa dovrà essere riferita nel testo tramite il comando `\ref{fig:etichetta}`.

3.2.7 Indice analitico

Il file “main” (per default `elaborato.tex`) contiene già i comandi necessari alla costruzione dell’indice analitico.

È tuttavia necessario inserire le varie voci di indice analitico. A tal fine in un qualunque punto del testo (ad esempio un paragrafo, una didascalia di una tabella ecc.) è possibile inserire una voce di indice analitico tramite il comando `\Index{voce}` oppure `\{voce!sottovoce}`. Un’unica avvertenza: è conveniente tenere traccia delle voci inserite su di un file testo separato e ordinato alfabeticamente. Infatti è facile dimenticare con il tempo le voci effettivamente impiegate inserite (ad esempio, si è usato il termine “norma” o “raccomandazione”? l’uso indifferente delle due precedenti produce due voci di indice analitico). In ogni caso, per quanto possibile, impiegare per le voci di indice analitico la forma singolare (come ad es. “progetto”, o “comando”) piuttosto che la forma plurale (come ad es. “progetti”).

3.2.8 Copertina

Il file `res/...tex` permette di definire (creare) una copertina.

Si invita il lettore a analizzare il file `guidaDocumentiEstesi.tex` ed eventualmente il file `res/sample-cover.input.tex`, i quali sono commentati in proposito.

Per brevità, per generare la copertina di questa guida si deve caricare la classe `gmeepd` con le opzioni adeguate. Sono inoltre disponibili comandi per la definizione di alcuni campi della copertina come autore, titolo ecc.

Un esempio è il seguente:

```
\documentclass[cover=work]{gmeepd}
...
\Author{M.Bertocco}
\Title{Una guida alla preparazione...}
\Subtitle{elaborato, tesi(na), libro,\ldots}
\Date{Dic.2008}
...
\begin{document}
\Cover % makes cover
...
\end{document}
```

Si noti che se nessuno dei tipi predefiniti risultasse soddisfacente, allora tramite il comando seguente:

```
\documentclass[cover=user,coverFile=...]{gmeepd}
```

si indica alla classe `gmeepd` di utilizzare il file specificato a lpосто dei puntini (nella directory `res/`) per la definizione della copertina.

3.3 Riservatezza del documento

Può capitare di non voler mostrare del testo in un documento a meno che non se ne voglia produrre una versione riservata. A tale fine si può far riferimento alle macro `\tSecret` e `\AltSecret`, al comando `\Secret` e all'opzione `mode` della classe `gmeepd`.

L'opzione `status=reserved`, indica che viene prodotta una versione "riservata" del documento, mentre l'opzione `status=normal` (default), indica che viene prodotta una versione non riservata del documento. Tale opzione determina quale testo viene mostrato o non mostrato, in accordo a quanto determinato dalle macro `\tSecret` e `\AltSecret`.

La macro `\tSecret` semplicemente mostra il testo definito come argomento se il documento era stato definito come riservato ossia si era attivata l'opzione `status=reserved` (`\documentclass[status=reseved]{gmeepd}`).

La macro `\tAltSecret` mostra il testo definito come primo argomento se il documento è definito come “riservato” (`status=reseved`) e altrimenti mostra il testo definito come secondo argomento. a titolo di esempio, tale macro è utile ad esempio per caricare due versioni diverse di una stessa tabella inserendo comandi del tipo `\input{...}` nello spazio relativo al testo pubblico e al testo riservato.

Per ottenere due versione diverse del documento è quindi necessario modificare l'opzione di caricamento della classe `gmeepd` (`status=reserved`, piuttosto che `status=normal` o nessuna opzione) e poi ricompilare.

Si può anche cambiare (temporaneamente) il grado di riservatezza di parte del testo, utilizzando la macro `\Secret`, la quale deve avere come parametro la stringa “true” oppure “true”. Tale comando altera lo stato del documento fino a quando non viene eseguita nuovamente la macro `\Secret` con parametro diverso.

Per sapere quale sia lo stato di riservatezza corrente si può verificare il valore della variabile globale `\GMSecret`, la quale è `true`, se lo stato del documento è “riservato”, e `false` altrimenti.

Infine, invocando la classe `gmeepd` con parametro `status=debug`, viene comunque mostrato il testo definito come “secret”, o entrambi le versione del testo definito dalla macro `\tAltSecret`, e però una nota a margine richiama il punto nel quale sono state invocate tali macro.

3.3.1 Un esempio

NOTA: Per vedere l'effetto delle macro suddette, è necessario ricompilare questo documento cambiando l'opzione di caricamento della classe `gmeepd`. Questo sottoparagrafo può risultare incomprensibile se non consulta il corrispondente “sorgente `LATEX`” e non si prova a cambiare il valore dell'opzione `status`.

Segue una frase con del testo che scompare se `\Secret` è `true`.

Parte pubblica, segue parte riservata

ancora parte pubblica, dopo la parte riservata (se non si è visto il testo “testo riservato”, allora questo documento è stato compilato con opzione `status=normal` oppure `status=normal` o ancora senza specificare l'opzione `status`.

fine paragrafo.

Un altro esempio: questa frase potrebbe contenere del testo segreto. Per vederlo il comando `\Secret` deve essere impostato a `false`. ora risulta `false`.

Questa frase contiene invece del testo che cambia a seconda del valore impostato per `Secret`.

Infatti il nome “Mario Rossi ” può essere riportato con un valore generico (o di fantasia) “pubblico“ piuttosto che con il nome vero e proprio che dovrebbe essere distribuito alle stampe in forma riservata. In questo esempio il nome segreto (!) è *James Bond*, mentre il nome sostitutivo per il pubblico generico è *Mario Rossi*.

Grado di riservatezza: questo documento è pubblico .

testo che vede chiunque (cioè quando `\Secret = false`) (Mario Rossi)

dir	contenuto/scopo
(root dir)	directory principale contiene i file \LaTeX relativi al testo, in particolare file il file master, eventuali script (unix/shell, windows/batch) ecc.
/bib	file BibTeX (.bib) per la definizione/inclusione di riferimenti bibliografici
/images	immagini (.eps) incluse tipicamente mediante le macro fornite dal file <code>rescmd.input.tex</code>
/tables	tabelle (file .tex inclusi mediante il comando <code>\input</code>)
/res	“risorse” ovvero file contenenti macro e/o comandi di impostazione generale, tipicamente i file di questa directory non vanno modificati
/source-images	file di appoggio dai quali vengono estratte le figure in formato .eps (ad esempio file .fig (generati da <code>xfig</code> , oppure file .ppt .odp dai quali tramite <code>OpenOffice.org</code> estrarre (esportare) le figure in formato .eps
/test	file di appoggio (test) che non fanno parte del progetto e usati per verificare parti di documento; ad esempio per raccogliere assieme tutte e sole le figure, le tabelle ecc.
/backup	eventuali copie di backup (.zip, .tgz ecc.) del lavoro
/template	file vuoti o modelli (da usare tramite un’operazione del tipo <code>copy&rename</code>)

Tabella 3.1: Struttura e contenuto delle directory

file	modif.	significato
<code>gmeepd.cls</code>		classe \LaTeX : macro, inclusione pacchetti, ecc.
<code>usercmd.input</code>	√	macro utente

Tabella 3.2: Elenco dei principali file `.tex` della directory principale e `res`; i files contrassegnati con il simbolo “√” sono intesi come modificabili (con attenzione, e producendo adeguate spiegazioni in un file testo allegato), di tutti gli altri $\tilde{\text{A}}$ caldamente sconsigliata la modifica.

nome comando	impiego
<code>\tBeginPar</code>	inizio paragrafo “sottolineato”
<code>\tDef</code>	definizione
<code>\tDefObj</code>	definizione - oggetto che viene definito
<code>\tDefTxt</code>	definizione - testo della definizione
<code>\tRemark</code>	frasi importanti (tipicamente in corsivo)
<code>\tREMARK</code>	frasi fondamentali (tipicamente grassetto corsivo)
<code>\tCode</code>	frammenti di codice
<code>\tForeign</code>	altra lingua (esempi: <i>operating system</i> , ...)
<code>\tLatin</code>	latino (esempi: <i>ex novo</i> , <i>de facto</i> , <i>id est</i> , ...)
<code>\tExample</code>	testo relativo a una breve frase inserita come esempio
<code>\tStandard</code>	citazione di un riferimento normativo
<code>\tQuote</code>	citazione (testo “copiato” da altro autore)
<code>\tQuoteFig</code>	riferimento a una figura o una sua parte
<code>\tConcept</code>	concetto da evidenziare

Tabella 3.3: nomi dei comandi (macro) che permettono l’adozione di uno stile del testo e loro impiego.

comando	carattere speciale	standard L ^A T _E X
<code>\qmo</code>	“ (virgolette aperte)	
<code>\qmc</code>	” (virgolette chiuse)	
<code>\par</code>	a capo (no spaziatura verticale)	√
<code>\</code>	a capo (con spaziatura verticale)	√
<code>\textbackslash</code>	\ (carattere backslash)	√
<code>\{</code>	{ (par. graffa aperta)	√
<code>\}</code>	} (par. graffa chiusa)	√
<code>\%</code>	% (car. percentuale)	√

Tabella 3.4: macro associate a caratteri speciali

comando (macro)	scopo
<code>\Fig</code>	inserimento di una figura in formato <code>eps</code>
<code>\FigPos</code>	come <code>\Fig</code> , inoltre permette di forzare il posizionamento della figura
<code>\FigScaled</code>	come <code>\Fig</code> , inoltre permette di definire un fattore di scala orizzontale
<code>\FigScaledLarge</code>	come <code>\Fig</code> , inoltre scala la figura in modo che sia larga quanto il testo
<code>\FigRotatedScaled</code>	come <code>\Fig</code> , inoltre ruota la figura di un quarto di giro e permette di scalarla
<code>\FigScaledPos</code>	come <code>\Fig</code> , inoltre permette di forzare il posizionamento della figura e definire un fattore di scala orizzontale
<code>\FigTwoVScaled</code>	inserisce due figure <code>eps</code> una sopra l'altra e permette di scalarle in larghezza
<code>\FigTwoHScaled</code>	isersisce due figure <code>eps</code> affiancate e permette di scalarle in larghezza

Tabella 3.5: comandi predefiniti (macro) utili all'inserimento di figure in formato `eps`: nomi e funzione svolta dal comando

comando (macro)	parametri
<code>\Fig</code>	<code>{eps-file} {short-caption} {caption} {label}</code>
<code>\FigPos</code>	<code>{position} {eps-file} {short-caption}</code> <code>{caption} {label}</code>
<code>\FigScaled</code>	<code>{position} {eps-file}</code> <code>{short-caption}{caption}{label}</code>
<code>\FigScaledLarge</code>	<code>{eps-file} {short-caption} {caption} {label}</code>
<code>\FigRotatedScaled</code>	<code>{eps-file} {short caption} {caption} {label}</code> <code>{height}</code>
<code>\FigScaledPos</code>	<code>{position} {eps-file} {short-caption}</code> <code>{caption} {label} {width} {position}</code>
<code>\FigTwoVScaled</code>	<code>{eps-file-top} {eps-file-bottom}</code> <code>{short-caption} {caption} {label}</code> <code>{width-top} {width-bottom}</code>
<code>\FigTwoHScaled</code>	<code>{eps-file-left} {eps-file-right}</code> <code>{short-caption} {caption} {label}</code> <code>{width-right} {width-left}</code>
parametro	significato
<code>eps-file</code>	nome di un file eps (senza estensione) della directory <code>images</code>
<code>short-caption</code>	descrizione (breve) della figura che dovrà comparire nell'elenco delle figure in fondo all'elaborato;
<code>caption</code>	didascalia;
<code>label</code>	definisce l'etichetta da dare alla figura (che verrà riferita nel testo tramite il comando <code>\ref{fig:label}</code>);
<code>position</code>	sono i caratteri di posizionamento delle figure (si veda una guida di <code>L^AT_EX</code>) del tipo <code>h</code> , <code>b</code> , <code>t</code> , <code>p</code> ;
<code>width...</code>	parametri di larghezza (anlogamente di altezza): possono essere riferiti in <code>inch</code> , <code>cm</code> punti (es. <code>2.54cm</code> , <code>1in</code> , <code>54pt</code>) o in termini relativi rispetto alla larghezza del testo (es. <code>.5\textwidth</code> , <code>.3\textheight</code>)

Tabella 3.6: sintassi dei comandi aggiuntivi (macro) utili all'inserimento di figure in formato eps.

opzione	contenuto
(1/2 immagini)	
shortcaption	didascalia breve (finisce nell'elenco delle figure)
label	(va richiamata mediante <code>\ref{fig:label}</code>)
pos	comandi di posizionamento (a scelta tra <code>hbp!</code>)
1 immagine	
file	file eps (nome senza ext), dir <code>images/</code>
width	larghezza immagine (es. <code>.5\textwidth</code>)
height	altezza immagine (es. <code>.5\textheight</code>)
angle	angolo di rotazione figura
rotate	true (ruota di 90 gradi la figura)
2 immagini	
mode=horiz	(=2 figure in orizzontale)
fileL, fileR	file immagini eps di sinistra e destra
widthL, widthR	fattori di scala orizz. immagine di Sn e Dx
heightL, heightR	fattori di scala vert. immagine di Sn e Dx
2 immagini	
mode=vert	(=2 figure in verticale)
fileUp, fileDw	file immagini eps di sopra e di sotto
widthUp, widthDw	fattori di scala orizz. immagine di sopra e di sotto
heightUp, heightDw	fattori di scala vert. immagine di sopra e di sotto

Tabella 3.7: Opzioni relative alla macro `\Figure`

nome (<code>\CoverType</code>)	tipo di copertina (parametri possibili)
<code>work</code>	copertina di default, utile per la compilazione di lavori di approfondimento, rapporti ecc. (Author, Title, [Subtitle, Date])
<code>thesis</code> (tesi)	tesi di laurea triennale, specialistica, magistrale (Author, Title, Course, Advisor, Date [Subtitle, CoAdvisor])
<code>tesina</code>	tesina (Author, Title, Course, Advisor [SubTitle, Date])
<code>phdthesis</code>	tesi di dottorato di ricerca (Author, Title, PhDCourse, Director, Advisor [Subtitle, CoAdvisor, Date])

Tabella 3.8: comandi associati ai vari tipi di copertina, utili alla definizione del tipo di copertina. In parentesi sono elencati i possibili comandi ulteriori che ha senso impiegare per il particolare tipo di copertina e che verranno utilizzati per la predisposizione della stessa, valori opzionali sono inoltre indicati tra parentesi quadre.

Capitolo 4

Tabù:

ovvero le cose assolutamente da
NON fare (e quelle invece da
fare al loro posto)

Introduzione

In questo capitolo vengono brevemente elencate alcune avvertenze relative a stili di edizione da evitare come la peste.

Prima si fornisce un breve elenco contenente le *cose da NON fare*, e per poi dare alcune motivazioni corrispondenti.

4.1 Elenco dei tabù

4.1.1 Uso della lingua italiana

La presente guida è rivolta a (allievi) ingegneri ai fini della preparazione di documenti tecnici. Per la natura di questi ultimi, non è strettamente necessario che il testo sia “avvincente”. Ciò premesso, si riportano alcuni consigli nella speranza che permettano all’autore di migliorare la qualità del testo che sta per essere sviluppato.

- ***fare uso della forma impersonale.*** In altri termini: sostituire le frasi del tipo “con questi passaggi dimostro che”, “ho trovato”, “ho costruito”, ... con corrispondenti frasi del tipo “i passaggi precedenti dimostrano che”, “si trova (si è trovato)”, “si è costruito”, ...

- *inserire citazioni bibliografiche!*

Ad esempio se viene richiamata una norma evitare frasi del tipo “le norme prescrivono ...” senza citare le norme stesse, a meno che non sia ben chiaro dal contesto quali siano le norme in questione. Ci si aspetta E dunque la frase precedente va riformulata come “le norme [...] prescrivono ...”, dove [...] è il riferimento bibliografico corrispondente (comando $\text{\LaTeX \cite{...}}$)

Analogamente, citare le fonti bibliografiche di teoremi, formule, schemi, algoritmi ecc. variamente impiegati nell’articolazione del ragionamento esposto.

- Il *grado di dettaglio* con il quale viene redatto il documento deve essere commisurato al lettore al quale il documento stesso è rivolto. Nel caso di una tesi (tesina, lavoro di approfondimento per un corso ecc.) *il lettore di riferimento è un compagno di corso che non sa nulla della tesi in questione (tesina o lavoro di approfondimento che sia)* e che però ha superato gli stessi esami dell’autore. In particolare, il lettore di riferimento e al quale il lavoro dovrebbe essere rivolto *non è il relatore (o docente del corso)*, il quale invece valuterà il lavoro stesso!

Tenendo in mente questa regola dovrebbe essere semplice capire a quale grado di dettaglio arrestare la descrizione.

- *progettare la stesura del documento* iniziando dall’indice. Poi espandere parti dello stesso indice “per punti” (come se fosse una presentazione) e infine scrivere il testo nel suo dettaglio. Ciò permette di analizzare sia il grado di coerenza del testo, sia e la consequenzialità delle argomentazioni prima ancora di aver scritto il documento in dettaglio.

In particolare, si consiglia di scrivere per primi i capitoli “centrali”, ovvero quelli che riportano i risultati o le deduzioni oggetto dell’elaborato (cioè *la tesi!*). Poi scrivere i capitoli che riportano le necessarie premesse ai risultati trovati (es. base di partenza, enunciati di teoremi importanti ai fini seguenti, risultati altrui migliorati nel corso del lavoro svolto). se tale lavoro viene svolto per secondo, è più facile determinare cosa scrivere in funzione delle informazioni che il lettore deve avere per poter poi comprendere quanto riportato poi nei capitoli “centrali”. Le varie introduzioni ai capitoli o il capitolo di introduttivo iniziale all’elaborato vanno redatti *per ultimi*.

4.1.2 Stili (tipografici)

Non usare i comandi \LaTeX per cambiare lo stile al testo (esempio tramite i comandi del tipo `\emph`, `\textit`,...). Usare invece delle macro riferite al motivo per il quale si vuole cambiare lo stile del testo (è una definizione?, una parte di testo importante?, una citazione?, ... allora per ciascuno dei casi precedenti ci si aspetta l'impiego di un comando opportuno).

A questo proposito sono già definiti parecchi stili (comandi): si veda la tabella 3.3 a pag.21.

Se proprio fosse necessaria una nuova “macro”, allora definirla nel file `res/ usercmd.input.tex`, traendo ispirazione da quelle già disponibili nel file `res/ cmd.input.tex`. *Non cambiare* il file `res/cmd.input.tex`¹.

4.1.3 Figure

- Nel caso di figure “a tratto” (come ad esempio uno schema a blocchi, o uno screenshot), possibilmente *impiegare formati vettoriali* (anzichè di tipo *raster*). Comunque, *non utilizzare MAI il formato jpeg*, perché tale formato è “lossy”, con il risultato che la figura (a stampa) assumerà una qualità sicuramente scadente (es. linee sbiadite, contorni mal definiti testo difficilmente leggibile). Se proprio si fa impiego di immagini raster, allora utilizzare al più; i formati GIF oppure PNG. Il formato jpeg produce buoni risultati solamente nel caso di fotografie (es. un paesaggio, un ritratto di una persona).
- Nella preparazione di una figura del tipo schema a blocchi, *usare per i font un corpo (num. di punti) “adeguato”*. La regola da seguire è che la dimensione del font sia coerente con quella del resto del testo. In buona sostanza, nella versione stampabile dell'elaborato la dimensione dei caratteri delle figure e dei caratteri del resto del testo dovrebbe essere apparire come approssimativamente uguale.

Se il vincolo precedente rende complicato realizzare una figura, allora suddividere la figura in più figure, eventualmente costruendone diverse tra loro collegate con grado di dettaglio crescente.

- Usare per le immagini i font “arial/helvetica” oppure “times”. Evitare altri tipi di font (forse coreografici, ma di difficile lettura) e soprat-

¹al momento della ricezione dell'elaborato le modifiche inserite in tale file non verranno recepite e poi l'elaborato verrà ricompilato per verifica usando invece la versione originale del file `cmd.input.tex` stesso; in caso di errore in compilazione l'intero elaborato potrebbe essere rifiutato (!)

tutto evitare assolutamente font “arabescati” o “da fumetti” come ad esempio il “gotico” o il “western”.

Inoltre, usare un solo tipo di font all’interno delle figure e per tutte le figure, per quanto possibile, usare sempre lo stesso font.

- *Rendere per quanto possibile la didascalia della figura auto-esplicativa*, a costo che quest’ultima diventi molto lunga, oppure al prezzo di ripetere cose già scritte nel resto del testo, o in un’altra didascalia.

Ad esempio, evitare frasi del tipo “Risultati sperimentali”: aggiungere invece il titolo dell’esperimento e riportare tutte le condizioni (numeriche) riferite all’esperimento corrispondente. O ancora una didascalia del tipo “Screenshot no.3” è praticamente inutile: vanno anche riportate le condizioni sotto alle quali quello screenshot è stato preso. In buona sostanza, il lettore non deve essere costretto a cercare nel resto del testo i dati che rendono chiaro a cosa una data figura si riferisca: la didascalia deve contenere a tal fine tutte le informazioni sufficienti alla comprensione della figura stessa.

- Evitare assolutamente di riferire nel testo figure (e tabelle) con frasi del tipo “la figura seguente...” o “nella figura di sopra...”, perché; \LaTeX impagina le figure (e le tabelle) spostandole di volta in volta sulla pagina più opportuna in accordo a regole tipografiche (il “sopra” diventa facilmente un “sotto” o un “due pagine oltre!”). Quindi nel riferire una figura nel testo, usare le cross-references (tag \LaTeX `ref{fig:...}` `ref{tab:...}`) e frasi del tipo “in figura x.y...” (“la tabella z.w...”)
- non modificare il file `gmeepd.cls`, se una macro “non piace”. Il file `res/usercmd.input.tex` è pensato invece per questo.
- non includere formati diversi da `eps` tramite comandi \LaTeX (anche se possibile!). La corrispondente importazione in forma automatica può richiedere programmi di appoggio dipendenti dalla piattaforma operativa (!).

Invece, creare *sempre* un file `eps` per la successiva inclusione.

- Usare per quanto possibile le macro definite nel file `res/cmd.input.tex`. E se proprio fosse necessario crearne di nuove, allora salvarle nel file `res/usercmd.input.tex`.
- salvare i “sorgenti” delle figure (ossia i file a partire dai quali i corrispondenti `eps` sono stati creati) nella directory `source-images`

(ad esempio la presentazione OpenOffice.org dalla quale sono stati esportati gli schemi a blocchi, i file in formato png creati da un'operazione di “cattura schermo”, ecc.

4.1.4 Tabelle

Per quanto riguarda le tabelle è difficile creare stili e comandi che permettano di automatizzare il processo di costruzione. Nella directory `template`, sono però disponibili alcuni esempi.

Alcune avvertenze di carattere generale:

- per quanto possibile (e cioè sempre!) *evitare le fincature verticali* : fanno solo confusione;
- separare l'intestazione della tabella dal corpo della tabella con una riga orizzontale (comando `\hline`);
- inserire una riga orizzontale in testa e in coda alla tabella: ciò separa chiaramente la tabella dal resto del testo e dalla didascalia;
- usare l'environment `\table` (oppure `\longtable`) per la creazione della tabella e inserire l'eventuale label all'interno del comando `\caption`; infine, per la label usare il formato `\label{tab:nomeTabella}`;

4.1.5 Bibliografia

Vale una regola fondamentale:

usare BibTeX.

Per approfondimenti circa BibTeX di veda [3], oppure [1, 2]. In breve:

- preparare uno o più file BibTeX (estensione `.bib`) contenenti i vari “record”, e salvarli nella directory `bib/`. Si veda a titolo di esempio il file `bib/bib-sample.bib`;
- per inserire un riferimento bibliografico usare il comando `\cite{}`; come label, impiegare quella definita nel file `bibtex!`;
- sempre a proposito di label si raccomanda di evitare “voli di fantasia” per la label da assegnare a un riferimento bibliografico perché i file `bibtex` nel tempo verranno condivisi da più persone: preferire invece un formato del tipo “CognomeAutore-anno-numero”, oppure

“CognomeAutore-rivista-anno-numero”; ad esempio: non usare label del tipo “ArticoloFico-1” oppure “TesiLele”, ma invece scegliere etichette del tipo “Einstein-1945”, o “Shannon-IEEEproc-1970”, o ancora “Rossi-tesiDEI-2007”;

- in fondo al file “main” (per default `elaborato.tex`), istruire \LaTeX a usare i file bib prodotti mediante il comando `\bibliography{file, file, ...}` (senza estensione, es. `\bibliography{bib/bib-latex, bib/bib-sample}`); e poi, se si vogliono inserire tutti i record contenuti nei file BibTeX indicati, allora invocare il comando `\cite{*}`, in caso contrario non inserire nulla;
- si ricorda che poi, da linea comandi è necessario eseguire: `latex; bibtex; latex; latex`; in modo da risolvere le cross reference, usare bibtex per generare la bibliografia e includerla finalmente nel documento a stampa.
- Da ultimo, si fa presente come vi siano dei tools per la conversione automatica da riferimenti bibliografici online a BibTeX. In buona sostanza, quando si consulta una riferimento bibliografico tramite accesso a banche dati come IEEEExplore, ISI web of science ecc. è possibile effettuare operazione di “copy and paste” dei link presenti nella banca dati verso form di convertitori on line i quali forniscono poi i record per BibTeX. Per approfondimenti in proposito si veda il link relativo a “BibConverter” [4], una demo è anche disponibile al link [5].

4.1.6 Cross-reference

Si ricorda che per generare una voce di cross-reference si impiega la coppia di comandi `\label{...}` e `\ref{...}`. Nasce l’esigenza di riferire con numeri progressivi diversi figure, tabelle, equazioni, capitoli, paragrafi. A tal fine si consiglia caldamente di impiegare per le etichette il formato “counterName:name”, dove “counterName” permette di distinguere tra il numero progressivo assegnato a una tabella rispetto a quello dato a una figura ecc., mentre “name” è l’etichetta vera e propria. Per quanto riguarda i nomi da impiegare per “counterName” si raccomandano quelli riportati in tabella 4.1.

Ad esempio la label associata alla tabella 4.1 è `tab:counterName`, mentre quella associata a questo capitolo (cioè il numero 4 con inizio a pag.27) è `chap:tabu`, e infine la label di questo paragrafo (4.1.6) è `sec:crossReference`.

counterName	impiego
<code>part</code>	parte
<code>chap</code>	capitolo
<code>sec</code>	paragrafo (section)
<code>tab</code>	tabella
<code>fig</code>	figura
<code>eq</code>	equazione

Tabella 4.1: Parole chiave da adottare per i principali contatori associati alle cross-reference. Il comando da usare per la definizione di una voce di cross reference è `\label{counterName:nome}`, mentre il numero di cross reference si ottiene tramite il comando `\ref{counterName:nome}`

Parte II

Esempi

Capitolo 5

Per chi ha fretta

Introduzione

Il presente capitolo è scritto per chi ha fretta ...

E quindi lo stile sarà conciso.

I comandi sono riferiti a un sistema operativo `unix/Linux` (e forse anche `windows`, quando ha senso spiegare le differenze).

5.1 Al lavoro!

5.1.1 Un primo test

1. Installare i vari package necessari per `LATEX` (vedi par.1.1.2 a pag.3).
2. Copiare il file `template-elaborato.zip` in una directory di proprio gusto. e decomprimerlo: tale operazione crea un subtree `elaborato/` contenente, tra gli altri, un file `main.tex`.
3. Provare subito a eseguire il comando `build.sh` (`build.bat`). Il risultato dovrebbe essere un file `main.pdf`, pronto per la visione e stampa. (Nel caso `unix` un secondo file `main.2p.pdf` contenente una versione dello stesso in formato “booklet”)

il risultato sarà un elaborato (tesi(na), libro,...) privo di contenuti utili, ma questo è un altro problema!

5.1.2 Modificare la copertina

Usare i comandi del tipo `\Author`, `\Title`, ecc. definiti all’inizio del file “master” (`main.tex`, per default). Non dovrebbe essere complicato...
E poi ricompilare.

5.1.3 Scrivere un nuovo capitolo

Copiare il file `template/void.tex` in uno nuovo (ad es `cap01.tex`), modificare il file `main.tex`, aggiungendo un comando per l’inclusione del file appena creato (`\include{cap01}`), e poi passare all’edizione del nuovo file (`cap01`) di conseguenza. Ad esempio, si può scrivere una frase, per poi passare alla generazione del file `pdf` corrispondente.

5.1.4 Per proseguire...

Se non si sono verificati problemi nelle prove precedenti, allora si può dichiarare concluso il primo test. Ovvero si può assumere \LaTeX installato e “funzionante” assieme ai tools correlati. Si è quindi in grado, almeno in linea di principio, di passare alla stesura dell’intero testo .

A tal fine Il paragrafo successivo tratta alcune esigenze diffuse (inserimento di figure,...) fornendo qualche dettaglio ulteriore. La lettura corrispondente può essere rinviata a un secondo momento sulla base di esigenze specifiche.

E’ però preliminarmente necessario comprendere la struttura data al documento nel suo complesso e come utilizzare la “macro” di \LaTeX ecc.

Quindi si prega vivamente il lettore di tornare al capitolo 2 a pag.7 e seguenti.

Infatti sono già disponibili parecchie “macro” utili per l’inserimento di figure o la definizione degli stili del testo.

E inoltre vi sono alcune avvertenze che è *necessario* seguire.

5.2 Come fare per...

5.2.1 Cambiare il nome “elaborato” ...

Ad esempio, se piace riferire il lavoro con altro nome (ad es. “tesi”) procedere come segue:

1. cambiare nome a `main.tex` (es. in `tesi.tex`)
2. modificare il file `build.sh` (`build.bat`) in modo che sia `tesi.tex` il documento di default, oppure invocare lo stesso tramite la riga `build.sh tesi`.

5.2.2 Non usare la linea comandi

Non è difficile: basta usare Kile (Linux) oppure TeXNicCenter (windows) o un prodotto similare, come ad esempio Led o Winedt:

unix usare kile

1. aprire un nuovo progetto, e dargli un nome (es. `miolibro`) (nella successiva finestra di dialogo evitare che Kile crei un nuovo file).
2. con il browser integrato in Kile aprire i file:
`main.tex`, `prefazione.tex` e `title.tex`;
quindi aggiungerli al progetto
3. tramite il menu Project/Project options specificare come “master document” il file `main.tex`

windows/WinEdt con WinEdt, aprire il file `main.tex` e poi tramite la voce di menu corrispondente impostare questo come master file

windows/TeXNicCenter aprire il file `main.tex`, tramite le icone è possibile avviare la compilazione. Si raccomanda di seguire gli step di compilazione LaTeX=>PS=>PDF.

5.2.3 Inserire una figura

Il modo di operare in dettaglio dipende dal tipo di figura (schema a blocchi, screencopy, fotografia,...).

In ogni caso tuttavia, si possono distinguere i seguenti passi fondamentali:

1. creare un file in formato `eps` contenente la figura
2. salvare il file `eps` nella directory `images`, e il corrispondente “sorgente” in `source-images`
3. usare le macro descritte al par.6.1

4. riferire la figura nel testo mediante il comando `\ref{fig:...}`

Si vede ora come generare un file `eps`, in alcuni casi fondamentali:

schema a blocchi Si assume lo schema a blocchi sia già disponibile in un file del tipo “presentazione”; e se non lo è conviene comunque usare un programma per preparare presentazioni (caldamente consigliato OpenOffice.org), perché così è più facile creare nuove figure modificando quelle già costruite

1. preparare la figura con un programma di disegno. Scelta consigliata: OpenOffice.org / Presentation, directory consigliata `source-images`. (se si usa `powerpoint`, si può eseguire `copy and paste` verso `openoffice ...`, in genere funziona abbastanza bene)
2. con, OpenOffice.org selezionare la porzione di figura che interessa da una slide, e poi selezionare la voce di menu `file/export`, quindi il formato `encapsulated postscript`, directory consigliata `images` (la scelta è obbligatoria, se si vogliono usare la macro predisposte nella classe `gmeepd`). Si vedano anche le figure 5.1 e 5.2

grafico (ottenuto tramite Matlab o programma consimile)

esportare il grafico i formato `eps`, poi usare le macro predisposte, come ad esempio `\Figure[...]{...}`.

screencopy

1. salvare la porzione di schermo di interesse con un programma di “cattura schermo”. *Non usare il formato `jpeg`!* (vanno bene i formati `png` o `gif`. Il formato `jpeg` non va bene, perché la compressione è *lossy*, e una volta riprodotta l’immagine appare sfuocata/confusa;
2. proseguire come nel caso delle fotografie.

fotografia

1. trasformare l'immagine (jpeg o equivalente) in formato `eps` e salvarla nella directory `images`. La conversione è facile. Basta usare il programma `convert`, il quale fa parte della suite `ImageMagick` (distribuita liberamente sia in formato eseguibile, sia in sorgente). Una linea comando tipica è `convert file.png file.eps`

In alternativa, aprire/creare una presentazione con OpenOffice.org; in una nuova pagina bianca importare l'immagine (menu `insert / picture / from file`); selezionare l'immagine appena importata e poi esportarla su file in formato `eps`, analogamente a quanto si fa nel caso di una figura del tipo “schema a blocchi”;

2. usare le macro predisposte nel file `res/cmd.input.tex`, come ad esempio `\Fig{...}` per inserire l'immagine in formato `eps`.

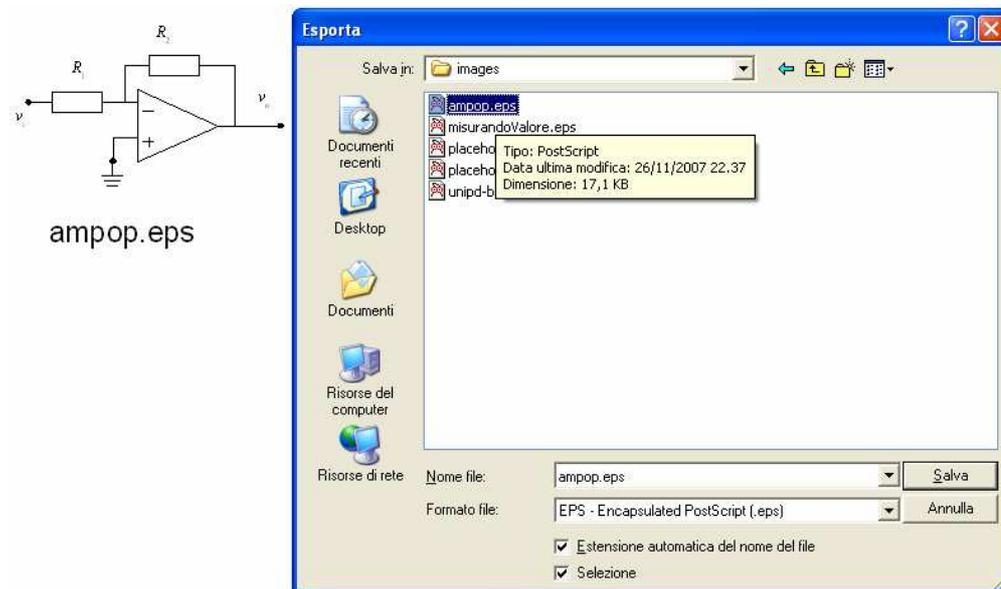


Figura 5.1: Esportazione di un'immagine eps tramite OpenOffice.org. Primo passo: selezione dell'immagine e del menu di esportazione.



Figura 5.2: Esportazione di un'immagine eps tramite OpenOffice.org. Secondo passo: scelta dei parametri di esportazione.

Capitolo 6

Esempi

Introduzione

In questo capitolo si mostrano alcuni esempi di impiego di stili e macro.

In prima lettura può essere saltato. Oppure semplicemente sfogliato per prendere atto del risultato ottenuto applicando i metodi descritti nella precedente parte contenente le buone regole da seguire per la preparazione del testo.

6.1 Figure

Gli esempi seguenti mostrano l'effetto dell'impiego dei comandi predefiniti per l'inserzione di figure (`eps`).

6.1.1 Impiego di comandi del pacchetto `\graphicx`

Di seguito vengono riportati alcuni frammenti di codice \LaTeX . Nel sorgente sono anche immediatamente ripetuti in forma esplicita in modo da includere la corrispondente figura.

Primo esempio: si veda la fig.6.1.

```
\begin{figure}[!h]
\begin{center}
\includegraphics{images/attention.eps}%
\caption[~esempio 1...]{inclusione dell'immagine
\tCode{attention.eps}}
\label{fig:attention}
```

```
\end{center}
\end{figure}
```



Figura 6.1: inclusione dell'immagine `attention.eps`

Secondo esempio: immagine ridimensionata (si veda la fig.6.2).

```
\begin{figure}[!h]
\begin{center}
\includegraphics[width=.25\textwidth]{images/attention.eps}%
\caption[~esempio 2...]{...}
\label{fig:attentionScaled}
\end{center}
\end{figure}
```



Figura 6.2: inclusione dell'immagine `attention.eps`, l'immagine viene scalata in modo che la sua dimensione orizzontale sia pari alla larghezza del testo moltiplicata per il fattore 0.12

6.1.2 macro `\Fig...`

Gli esempi seguenti mostrano l'effetto dell'impiego dei comandi predefiniti per l'inserzione di figure (`eps`).

In particolare la tabella 6.1, mostra la corrispondenza tra comando (macro), e numero di figura.

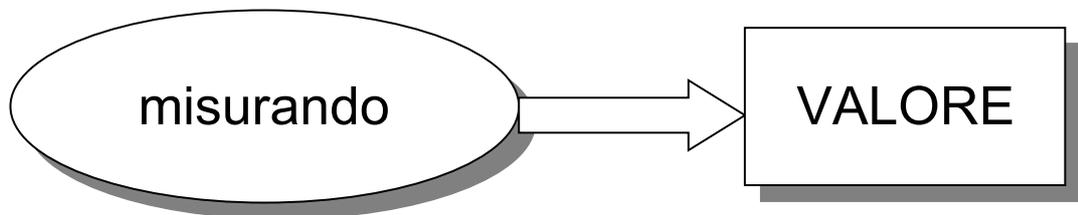


Figura 6.3: Esempio di impiego della macro `\Fig`, si noti che la figura è più larga del testo, inoltre la dimensione dei font è esagerata (*errori!*).

no. figura	comando
6.10	<code>\Fig</code>
6.11	<code>\FigPos</code>
6.12	<code>\FigScaled</code>
6.6	<code>\FigScaledLarge</code>
6.13	<code>\FigRotatedScaled</code>
6.14	<code>\FigTwoHScaled</code>
6.15	<code>\FigTwoVScaled</code>

Tabella 6.1: corrispondenza tra numero di figura e comando (macro) impiegato per l'inserimento della figura stessa.

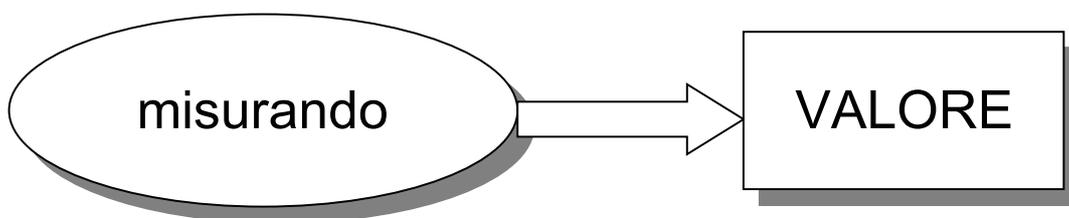


Figura 6.4: Esempio di impiego della macro `\FigPos`, l'effetto si comprende provando a cambiare il parametro 'position' usando le parole chiave 'h' (here), 'b' (page bottom), 't' (page top), 'p' (separate picture page (si veda la guida di riferimento di \LaTeX per il significato dei simboli precedenti, a proposito del comando `includegraphics`)).

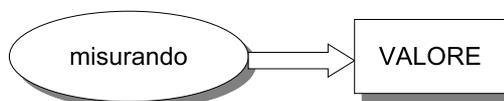


Figura 6.5: Esempio di impiego della macro `\FigScaled`, dove il fattore di scala è ipostato pari a `.5\textwidth` in questo caso la dimensione dei font che costituiscono la figura è coerente con quella del resto del testo.

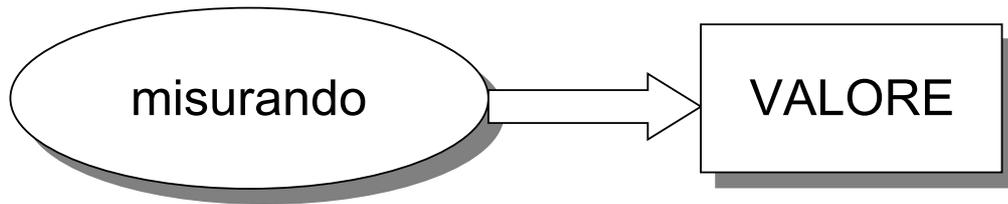


Figura 6.6: Esempio di impiego della macro `\FigScaledLarge`, tale comando è equivalente `\FigScaled` a con fattore di scala pari a `\textwidth`. Ancora una volta la dimensione dei font è esagerata.

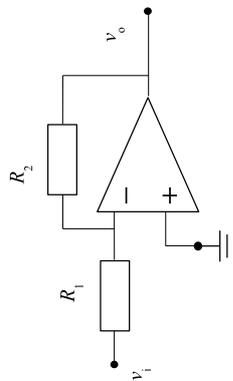


Figura 6.7: Esempio di impiego della macro `\FigRotatedScaled`, questa volta il fattore di scala scelto è pari a `.25\texttheight`.

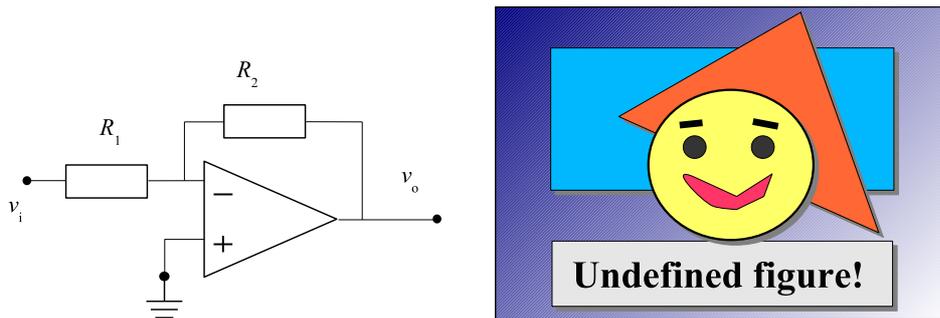


Figura 6.8: Esempio di impiego della macro `\FigTwoHScaled`, per le due figure è stato scelto lo stesso fattore di scala (`.45\textwidth`).

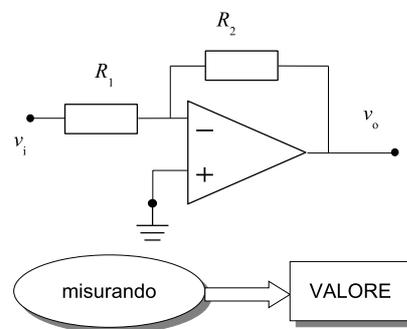


Figura 6.9: Esempio di impiego della macro `\FigTwoHScaled`, per le due figure è stato scelto lo stesso fattore di scala (`.4\textwidth`).

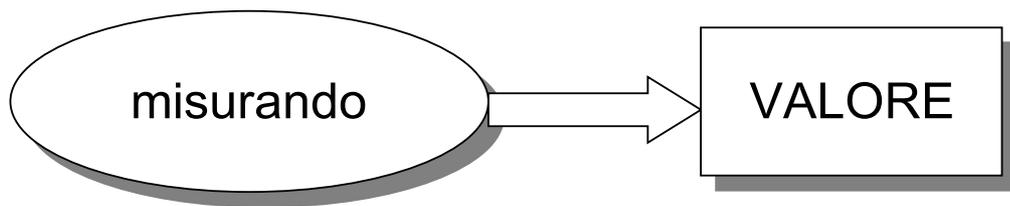


Figura 6.10: Esempio di impiego della macro `\Figure`, si noti che la figura è stata scalata alla larghezza del testo, inoltre la dimensione dei font è esagerata (*non va bene!*). La macro è stata invocata specificando i seguenti parametri: `\Figure[file=misurandoValore, shortcaption={macro \tCode{Fig}}, label=Fig]{caption}`.

6.1.3 macro `\Figure`

La tabella 6.2, mostra la corrispondenza tra comando (macro), e numero di figura.

no. figura	comando
6.10	<code>\Figure</code>
6.11	<code>\Figure pos=...</code>
6.12	<code>\Figure width=...</code>
6.13	<code>\Figure rotate=true, height=...</code>
6.14	<code>\Figure mode=vert</code>
6.15	<code>\Figure mode=horiz</code>

Tabella 6.2: corrispondenza tra numero di figura e comando (macro) impiegato per l'inserimento della figura stessa.

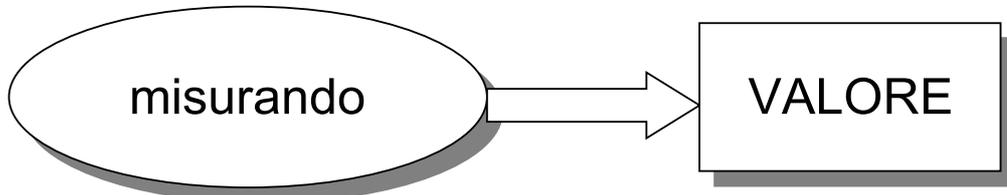


Figura 6.11: Secondo esempio di impiego della macro `\Figure`, l'effetto si comprende provando a cambiare il parametro 'pos' usando le parole chiave 'h' (here), 'b' (page bottom), 't' (page top), 'p' (separate picture page (si veda la guida di riferimento di \LaTeX per il significato dei simboli precedenti, a proposito del comando `includegraphics`)). La macro è stata invocata specificando i seguenti parametri: `\Figure[file=misurandoValore,pos=ht!,shortcaption={...},label=FigPos]{caption}`.

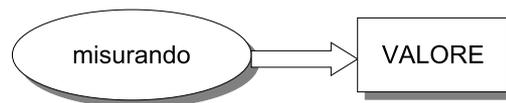


Figura 6.12: Esempio di impiego della macro `\FigScaled`, dove il fattore di scala è impostato pari a `.5\textwidth` in questo caso la dimensione dei font che costituiscono la figura è coerente con quella del resto del testo. La macro è stata invocata specificando i seguenti parametri: `\Figure[file=misurandoValore,width=.5\textwidth,shortcaption={...},label=FigScaled]{caption}`.

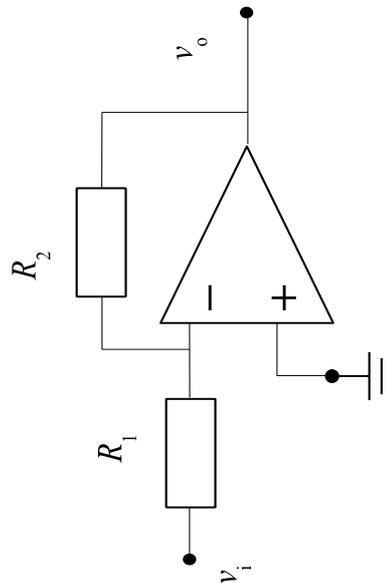


Figura 6.13: Esempio di impiego della macro `\Figure`, questa volta il fattore di scala scelto è pari a `.25\texttheight`, e la figura è ruotata (`rotate=true`). La macro è stata invocata specificando i seguenti parametri: `\Figure[file=ampop, rotate=true, height=.25\texttheight, shortcaption={...}, label=FigRotatedScaled]{caption}`.

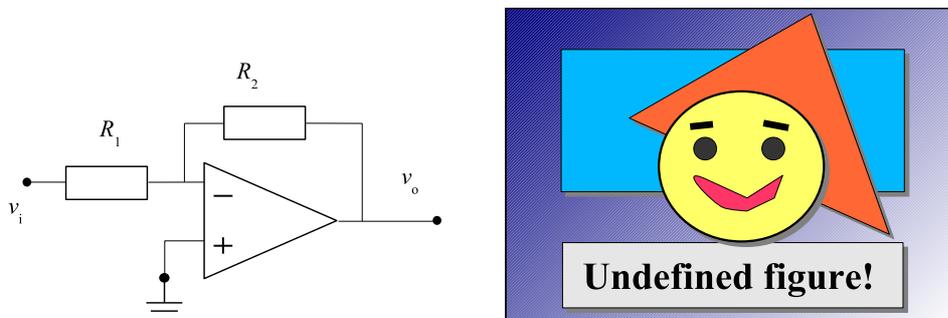


Figura 6.14: Esempio di impiego della macro `\Figure`, per le due figure è stato scelto lo stesso fattore di scala (`.45\textwidth`) e le due figure sono disposte affiancate (`mode=horiz`). La macro è stata invocata specificando i seguenti parametri: `\Figure[shortcaption={...}, label=FigPos]{caption}`.

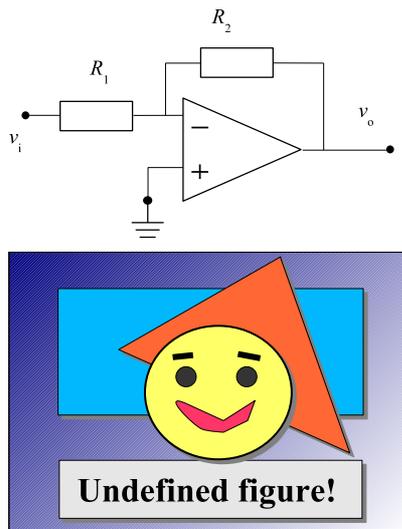


Figura 6.15: Esempio di impiego della macro `\Figure`, per le due figure è stato scelto lo stesso fattore di scala (`.4\textwidth`) e le figure sono disposte verticalmente (`mode=vert`). La macro è stata invocata specificando i seguenti parametri: `\Figure[mode=vert, fileUp=ampop, fileDw=placeholder, widthUp=.4\textwidth, widthDw=.4\textwidth, shortcaption={...}, label=...]{caption}`.

Bibliografia

- [1] A. Baudoin, “Impara L^AT_EX!” documentazione online di Latex, 1998.
- [2] T. Oetiker, “The not so short introduction to L^AT_EX 2_ε.” documentazione online di Latex, 1999.
- [3] L. Lamport, *LaTeX: A Document Preparation System*. Addison Wesley Professional, 2nd ed., June 1994. ISBN: 0-201-52983-1.
- [4] “BibConverter – a tool for converting citations.” Online available <http://www.bibconverter.net/>, 2007.
- [5] “BibConverter – a tool for converting citations: demo page.” Online available <http://www.bibconverter.net/demo/bmdemo.htm>, 2007.

Elenco delle tabelle

3.1	Struttura e contenuto delle directory	20
3.2	file res	20
3.3	macro per gli stili e impiego	21
3.4	caratteri speciali	21
3.5	figure: nomi macro e significato	22
3.6	figure: sintassi comandi aggiuntivi	23
3.7	opzioni \Figi	24
3.8	definizione copertina	25
4.1	nomi da dare ai contatori	33
6.1	figure di esempio	46
6.2	figure di esempio	49

Elenco delle figure

5.1	esportazione eps: step 1	42
5.2	esportazione eps: step 2	42
6.1	esempio 1: impiego di <code>\includegraphics</code>	44
6.2	esempio 2: immagine scalata	44
6.3	macro <code>Fig</code>	45
6.4	macro <code>FigPos</code>	46
6.5	macro <code>FigScaled</code>	46
6.6	macro <code>FigScaledLarge</code>	47
6.7	macro <code>FigRotatedScaled</code>	47
6.8	macro <code>FigTwoHScaled</code>	47
6.9	macro <code>FigTwoHScaled</code>	48
6.10	macro <code>Fig</code>	49
6.11	Figure: secondo esempio (pos)	50
6.12	Figure: terzo esempio(width)	50
6.13	Figure: rotate=true	51
6.14	Due figure in verticale	51
6.15	Due figure in verticale	52

Indice analitico

bibliografia	31	Figure	13
bibtex	31	graphicx	43
caratteri speciali	13	Secret	17
comando	12	stile	12
copertina	16, 25	tAltSecret	17
cross-reference	32	tSecret	17
directory		macro:Fig	14
res	20	macro:Figure	14
tree	11	organizzazione	11
documentazione	4	prefazione	III
figura	13, 39	prerequisiti	3
\Fig...	44	status	
\Figure	49	debug	18
eps	42	reserved	17
esempi	43	stile	12, 21
fotografia	41	tabella	31
grafico	40	template	11
inserimento	39		
schema a blocchi	40		
screencopy	40		
indice analitico	16		
installazione	3		
Kile	39		
label	33		
linea comandi	39		
macro	12, 21, 25		
\Fig...	44		
\Figure	49		
caratteri speciali	13		