

GRUPPO ITALIANO UTILIZZATORI DI T_EX

GUIDA D'USO ALLA CLASSE
GUIDATEMATICA



VERSIONE V.2.0.01 DEL 2020-04-29

Associati anche tu al GIT

[Fai click per associarti](#)

L'associazione per la diffusione di TEX in Italia, riconosciuta ufficialmente in ambito internazionale, si sostiene *unicamente* con le quote sociali.

Se anche tu trovi che questa guida tematica gratuita ti sia stata utile, il mezzo principale per ringraziare gli autori è diventare socio.

Divenendo soci si ricevono gratuitamente:

- l'abbonamento alla rivista $\text{A}\text{r}\text{s}\text{T}\text{E}\text{X}\text{nica}$;
- il DVD TEX Collection;
- un eventuale oggetto legato alle attività del GIT .

L'adesione al GIT prevede un quota associativa compresa tra 12,00 € e 70,00 € a seconda della tipologia di adesione prescelta e ha validità per l'anno solare in corso.

Guida d'uso alla classe *GUIDATEMATICA*

Copyright © 2016–2020, Gruppo Italiano Utilizzatori di TEX

Questa documentazione è soggetta alla licenza LPPL ($\text{L}\text{A}\text{T}\text{E}\text{X}$ Project Public Licence), versione 1.3 o successive; il testo della licenza è sempre contenuto in qualunque distribuzione del sistema TEX e nel sito <http://www.latex-project.org/lppl.txt>.

Questo documento è curato da Gruppo Italiano Utilizzatori di TEX .

INTRODUZIONE

Questa breve guida serve per descrivere la classe *guidatematica* predisposta per comporre le guide tematiche del $\text{\textcolor{teal}{G}\text{\textcolor{teal}{I}\text{\textcolor{teal}{T}}}}$ con uno stile comune, in modo che già il solo vedere il frontespizio permetta di riconoscere le guide tematiche dalle altre guide e dalle altre pubblicazioni del $\text{\textcolor{teal}{G}\text{\textcolor{teal}{I}\text{\textcolor{teal}{T}}}}$.

La classe *guidatematica* è dotata del file `.dtx` e la sua documentazione è ampliata con questo documento che ne descrive l'uso in modo più esteso. Sarebbe meglio che nessuno modificasse il file, altrimenti si perderebbe lo scopo per il quale è stato scritto, cioè di rendere omogenee nello stile le guide tematiche del $\text{\textcolor{teal}{G}\text{\textcolor{teal}{I}\text{\textcolor{teal}{T}}}}$. Se si desiderano proporre estensioni, cambiamenti, altre funzionalità, ci si deve rivolgere ai responsabili del $\text{\textcolor{teal}{G}\text{\textcolor{teal}{I}\text{\textcolor{teal}{T}}}}$ che curano il sito Web e le pubblicazioni del Gruppo.

Questo stesso documento è stato composto con la classe *guidatematica* in modalità `twoside`, che è quella predefinita.

Questa classe è costruita sulla classe *memoir* quindi l'autore può fare riferimento a tutte le funzionalità di quella classe, tranne alcune che collidono con i pacchetti già caricati; se decide di comporre l'indice analitico legga con attenzione l'apposito capitolo (vedi pag. 42); se decide di modificare lo stile di alcune liste o di crearne di nuove faccia riferimento alle funzionalità del pacchetto *enumitem* e alla sua documentazione.

Questa documentazione si riferisce alla classe *guidatematica.cls*, versione v.2.1.00, del 2020-04-29.

Essa è stata composta con $\text{\textcolor{teal}{L}\text{\textcolor{teal}{u}\text{\textcolor{teal}{a}}}\text{\textcolor{teal}{T}}\text{\textcolor{teal}{E}}\text{\textcolor{teal}{X}}}$ usando i font Latin Modern OpenType forniti insieme alla distribuzione di $\text{\textcolor{teal}{T}\text{\textcolor{teal}{e}}\text{\textcolor{teal}{X}}}$ Live fin dal 2012 e accessibili con i comandi propri di $\text{\textcolor{teal}{L}\text{\textcolor{teal}{u}\text{\textcolor{teal}{a}}}\text{\textcolor{teal}{T}}\text{\textcolor{teal}{E}}\text{\textcolor{teal}{X}}}$.

RINGRAZIAMENTI

Io ho scritto materialmente questa classe *guidatematica* cercando di mettermi nella veste del curatore di un lavoro collettivo del [GJIT](#) e per il [GJIT](#); ho consultato quindi gli autori o coautori delle guide tematiche pubblicate finora, Claudio Fiandrino, Roberto Giacomelli, Tommaso Gordini, Agostino De Marco, Mosè Giordano, Orlando Jovino, e ho seguito i suggerimenti di Ivan Valbusa; ho corretto alcune cose segnalatemi da Filippo Vomiero. Orlando Iovino ha inoltre contribuito alla creazione delle procedure bash (Linux) o bat (Windows) per installare tutto il materiale sul proprio PC.

Ringrazio tutti moltissimo del loro contributo, delle loro osservazioni, dei loro suggerimenti; questo è il miglior modo per contribuire al [GJIT](#), e quindi siamo tutti autori a pari merito di questa classe; se ne aggiungeranno altri col tempo e verranno via via aggiunti a questa lista.

Speriamo tutti che questo lavoro collettivo prosegua anche e soprattutto per comporre guide tematiche attraenti, chiare e, specialmente, utili. Questa classe riguarda prevalentemente l'estetica delle guide; il contenuto spetta agli autori, e forse una migliore veste grafica li invoglia a produrre altre guide. I frequentatori del Forum [GJIT](#) ne hanno bisogno.

CLAUDIO BECCARI

claudio dot beccari at gmail dot com

INDICE

INTRODUZIONE	3
RINGRAZIAMENTI	4
1 GUIDA RAPIDA ALLA CLASSE	7
1.1 Installazione della classe	7
1.2 Uso della classe	8
2 USO DELLA CLASSE	22
2.1 Pacchetti precaricati	23
2.2 Nuovi comandi	27
2.3 Nuovi ambienti	34
2.4 La virgola intelligente	39
3 GLI INDICI ANALITICI	42
4 pdfLaTeX, XeLaTeX o LuaLaTeX?	45
INDICE ANALITICO	50

GUIDA RAPIDA ALLA CLASSE

1

1.1 INSTALLAZIONE DELLA CLASSE

La classe viene distribuita in un kit da installare ed usare come quello distribuito per $\text{\texttt{Ar\TeX}}$ nic.

La via più spiccia è quella di copiare il kit `guidatematica.zip` (scaricabile dal link specificato alla pagina del sito [G_{IT} https://www.guitex.org/home/it/guide-tematiche/la-filosofia-delle-guide-a-tema](https://www.guitex.org/home/it/guide-tematiche/la-filosofia-delle-guide-a-tema)) nella cartella dove risiede il file sorgente (unico o principale) con cui si vuole comporre la propria guida tematica; in quella cartella si decompone il kit e tutti i file, compresi quelli della documentazione, risiedono in quella stessa cartella e sono immediatamente usabili per la composizione *solo* di quella guida tematica.

Meglio sarebbe disporre dell'albero personale di $\text{\texttt{T\TeX}}$; se non se ne dispone, lo si crea secondo le modalità descritte per il proprio sistema operativo e la propria distribuzione del sistema $\text{\texttt{T\TeX}}$. Se se ne dispone già, basta creare le cartelle a cui si fa riferimento qui di seguito. Si decompone il kit in una cartella qualsiasi, meglio se in¹ `$HOME/texmf/source/latex/guidatematica/`. Poi si spostano i file (eccetto quelli con l'estensione `.bst` e `.ist`) `GuidaTematica-doc.pdf` e `guidatematica.pdf` nella cartella `$HOME/texmf/doc/latex/guidatematica/` e gli altri file in `$HOME/texmf/tex/latex/guidatematica/`; il file `guidatematica.dtx` deve restare nella cartella dove si è eseguita la decompressione. Il file per lo stile bibliografico con estensione `.bst` va messo in `$HOME/texmf/bibtex/`

¹`$HOME` indica la propria cartella che generalmente ha il nome dell'autore, ma dipende molto dal sistema operativo; qui si è usato per il separatore delle cartelle la barra diritta, come è normale per le macchine di tipo UNIX; per le macchine Windows basta sostituire la barra diritta con la barra rovescia.

[bst/guidatematica/](#); i file per l'indice analitico, con l'estensione `.ist` vanno spostati in [\\$HOME/texmf/makeindex/guidatematica/](#).

Una terza possibilità consiste nel creare una propria cartella dove salvare il kit e dove decomprimere il tutto; non si sposta niente nelle cartelle del proprio albero personale, nemmeno in quelle che bisogna creare apposta; In queste cartelle si mettono invece i link simbolici ai file veri conservati nella propria cartella. In questo modo, quando si scarica un aggiornamento del kit, e lo si decompone, i link simbolici continuano a funzionare perché puntano a file con lo stesso nome, e non c'è bisogno di spostare niente. I link simbolici sono disponibili sulle macchine di tipo UNIX (Linux, Mac,...) e, sebbene con una sintassi diversa per crearli, anche su Windows dalla versione 8 in poi.

Se la distribuzione del sistema \TeX lo richiede anche per l'albero personale, bisogna ricordarsi di creare o di aggiornare il database dei nomi dei file di questo albero usando gli strumenti della propria distribuzione. A questo punto tutti i programmi di composizione del sistema \TeX sono in grado di trovare i file necessari per comporre *qualsiasi* guida tematica, qualunque sia la cartella dove i suoi file sorgente risiedono.

1.2 USO DELLA CLASSE

Lo schema generale da usare per comporre una guida tematica è il seguente:

```
\documentclass[b5paper,11pt]{guidatematica}
\ProvidesFile{⟨nome del file⟩}[⟨aaaa/mm/gg⟩ v.⟨x.y.z⟩ ⟨Descrizione⟩]

⟨qualche definizione o qualche ulteriore pacchetto⟩
⟨eventuali impostazioni per \makeindex⟩

\begin{document}

\def\authorspace{⟨spazio fra più autori⟩}
\author{⟨autore o autori⟩}
\title{⟨titolo⟩}
⟨eventuale dichiarazione di licenza diversa dalla LPPL⟩
\GetFileInfo{\jobname.tex}
```


1.2. USO DELLA CLASSE

```
\maketitle

\chapter{\langle titolo dell'introduzione \rangle}
\langle corpo dell'introduzione \rangle
\newpage
\tableofcontents
\langle eventuali \listoffigures e/o \listoftables con o senza asterichi \rangle
\clearpage

\mainmatter*

\chapter{\langle titolo del capitolo \rangle}
\langle corpo del capitolo \rangle

...

\backmatter
\chapter{\langle titolo della "postfazione" \rangle}
\langle corpo dell'eventuale postfazione \rangle

\langle eventuale ridefinizione di \prebibhook per fare precedere un breve testo
esplicativo all'elenco bibliografico vero e proprio \rangle

\bibliography{\langle nome del database bibliografico senza estensione \rangle}

%%% eventuali indici analitici
\printindex
\printindex[\langle nome dell'indice \rangle]
...
\end{document}
```

OSSERVAZIONI

Il testo precedente, benché scritto in un file PDF, può essere selezionato, copiato e incollato in un file `.tex` e usato come modello (o come *template* come è venuto di moda chiamarlo) per iniziare a strutturare il file sorgente di una guida tematica.

Comunque vale la pena di illustrare alcuni punti. Nel farlo, ci si rivolge a persone che, avendo qualcosa da dire in merito a qualche sfaccettatura del “sistema T_EX”, hanno sicuramente esperienza nello scrivere documenti usando il mark-up L^AT_EX, e conoscono molto bene pregi e difetti dei programmi di composizione pdfLaTeX, XeLaTeX e LuaLaTeX. Quindi molte cose saranno date per scontate.

OPZIONI DELLA CLASSE

Per impostazione predefinita della sottostante classe *memoir*, la composizione viene eseguita per la stampa fronte retro. L’autore può specificare l’opzione *oneside*; dato lo stile delle pagine simmetrico, le uniche differenze che si manifestano con questa opzione sono che (a) le note marginali appaiono sempre nel margine di destra e che (b) non vengono lasciate pagine bianche per aprire i capitoli solo sulle pagine dispari; stampando fronte retro, questo fatto potrebbe essere un inconveniente, ma nello stesso tempo consente di ridurre un poco la quantità di carta usata. Nello stesso tempo lo stile delle pagine perfettamente simmetrico permette di leggere la guida composta anche su schermo, senza avere quel noioso spostamento alternato dello specchio di stampa a destra e a sinistra. Per questo motivo si suggerisce di non comporre con l’opzione *oneside*. Invece, in vista della lettura a schermo, ma anche per avere una versione stampata (e rifilata), si suggerisce di comporre con l’opzione *b5paper* che è l’opzione predefinita; le altre opzioni per il formato della pagina sono *a4paper*, *a5paper*, *pad*, *tablet*; quest’ultima, poco raccomandabile, richiede molto lavoro per conferire un aspetto decente alla piccola pagina di 90 mm per 120 mm. Probabilmente lo specchio di stampa scelto per questo formato è troppo piccolo e non consente l’uso di certi pacchetti per mostrare il codice affiancato dal suo esito compositivo; tuttavia se si vuole usare, questo formato è leggibile su un piccolo tablet durante gli spostamenti sui mezzi pubblici. L’opzione *pad* serve per comporre su uno schermo da iPad, o simili apparecchi, supponendo una pagina virtuale di 120 mm per 160 mm. La lettura è più agevole, ma richiede comunque una buona dose di lavoro per evitare un numero eccessivo di righe sporgenti dai margini; queste sono relativamente frequenti anche con i formati maggiori e si manifestano quando il testo contiene lunghi indirizzi Web, oppure codice in modo verbatim. Per l’opzione *ipertesto* (non più disponibile) si veda il punto seguente.

1.2. USO DELLA CLASSE

PREAMBOLO

Il preambolo in generale può essere vuoto; può succedere però che l'autore necessiti di qualche definizione particolare che non si trova in nessun pacchetto o che abbia bisogno di ulteriori pacchetti oltre quelli già caricati (si veda più avanti l'elenco di questi pacchetti). In ogni caso, se desidera comporre la sua guida tematica in modo da usare collegamenti ipertestuali, con le versioni precedenti di questa classe, l'autore deve egli stesso specificare l'opzione *ipertesto* fra le opzioni; questa opzione ritardava al momento di completare il comando `\begin{document}` la chiamata del pacchetto *hyperref* con alcune personalizzazioni; il pacchetto veniva caricato per ultimo perché questa operazione deve essere fatta dopo il caricamento dei pacchetti personali. D'altronde i collegamenti ipertestuali non sono obbligatori, ma sono utili per la lettura sul monitor solo per alcune guide e non per altre; è una scelta dell'autore quella di voler usare i collegamenti ipertestuali ed è quindi una sua scelta personale quella di specificare l'opzione suddetta. Il guaio è che alcuni pacchetti recenti, come *glossaries*, richiedono di essere caricati dopo *hyperref*. Perciò si è tolta dalle funzionalità di questa classe l'opzione *ipertesto*, e si lascia all'utente la responsabilità di caricare i pacchetti che gli servono, compreso *hyperref*, nell'ordine giusto. All'inizio del corpo del documento viene eseguito in test per verificare se *hyperref* sia stato caricato e, nel caso, ne imposta alcune opzioni.

IL MATERIALE INIZIALE

Il comando `\frontmatter*` si comporta come il normale `\frontmatter` salvo che non esegue cambi di stile alla numerazione delle pagine, e quindi la numerazione comincia da 1 in cifre arabe e prosegue con le cifre arabe che non vengono reimpostate con `\mainmatter*`. Ma, come con il comando `\frontmatter` standard, i capitoli e gli altri comandi di sezionamento producono sezioni non numerate, che però vengono inserite nell'indice generale. I comandi di sezionamento asteriscati, invece, continuano a non comparire nell'indice.

Se l'autore volesse mantenere la numerazione delle pagine diversa per la front matter rispetto alla main matter, può tranquillamente usare i corrispondenti comandi non asteriscati; in quel caso la parte iniziale viene numerata con la numerazione romana composta in maiuscolo; eventuali voci della parte iniziale che debbano comparire nell'indice generale o in quello analitico vengono riportati comunque con i numeri coerentemente

composti in maiuscoletto. Non entro nel merito dell’opportunità della numerazione romana nella parte iniziale; questa è unicamente una scelta stilistica dell’autore della guida tematica. Ho solo provveduto a far sì che, qualunque decisione prenda l’autore, egli non deve fare acrobazie per rendere coerenti le numerazioni nei riferimenti vari che possono comparire nella guida.

PAGINA DEL TITOLO

La pagina del titolo e il retrofrontespizio vengono generati in automatico con i pochi dati forniti mediante le istruzioni `\author`, `\title`, eventualmente `\subtitle` e le eventuali dichiarazioni per specificare una licenza diversa da quella preimpostata mediante il comando `\licenza{⟨dichiarazione⟩}`. Si suggerisce di inserire anche la riga `\GetFileInfo` con il suo argomento, in modo da recuperare dal file sorgente le informazioni della versione e della data che, se disponibili, verranno indicate in calce al frontespizio. Per il comando `\licenza` si veda più avanti.

In generale si legge nelle guide che le informazioni per il frontespizio inserite con i comandi `\author`, `\title` e `\date` vanno inseriti nel preambolo e solo il comando `\maketitle` va inserito dopo `\begin{document}`. Nel caso delle guide tematiche, invece, anche quelle informazioni vanno inserite *dopo* `\begin{document}` e, ovviamente, prima di `\maketitle` per la maniera in cui sono differite alcune definizioni. Il comando `\date` va evitato, perché le informazioni corrette vengono recuperate dal file sorgente al quale sia stato inserito il comando `\ProvidesFile` con i suoi argomenti, come specificato sopra. È ovvio che senza questa istruzione il comando `\GetFileInfo` non può recuperare nessuna informazione. Trattandosi di “guide”, invece, queste informazioni sono essenziali.

Va notato invece il comando `\authorspace` da usare fra i nome degli autori per separarli di quanto specificato con *⟨spazio fra pi u autori⟩*; è evidente che se l’autore è uno solo, non occorre fare nulla; ma se gli autori sono più di uno fra le generalità di ognuno va inserito un comando di spaziatura, `\authorspace`; questo è predefinito per essere soltanto uno spazio; facendo precedere l’indicazione di `\author` con una diverse definizione con `\def` o con `\newcommand` e con `\let`, i vari nomi vengono composti spaziati; ognuna delle seguenti definizioni (con spazi diversi) è altrettanto valida:

```
\def\authorspace{\hspace{1.5em}}
```

1.2. USO DELLA CLASSE

```
\newcommand*\authorspace{\quad}  
\let\authorspace\qqquad
```

Nello scrivere i nomi dell'unico o dei vari autori si abbia l'accortezza di indicare sempre i nomi propri senza abbreviazioni prima dei corrispondenti cognomi. In caso di nomi e cognomi molto lunghi invece di uno spazio orizzontale si può usare anche un comando di nuova riga.

ISCRIZIONE AL G_{IT}

L'invito a iscriversi al G_{IT} viene stampato in automatico nel retro del frontespizio; esso consiste in una pagina a se stante con l'invito e con l'URL del sito per procedere all'iscrizione. La pagina appare come in questa guida sul verso della pagina del frontespizio. Se si usa il pacchetto *hyperref* l'URL è attivo e cliccandoci sopra si viene portati alla pagina in questione.

Sempre in automatico, in calce a questa pagina viene indicata la licenza a cui è sottoposta la guida tematica; se l'autore non interviene mediante il comando `\licenza`, viene dichiarato che la guida è sottoposta alla LPPL (L^AT_EX Project Public Licence, versione 1.3 o successive); altrimenti viene composta la dichiarazione dell'autore eseguita tramite il comando `\licenza` suddetto (vedi pagina 17).

INTRODUZIONE

Questo capitolo non numerato scritto nella “front matter” può avere un titolo diverso da “Introduzione”. Non è nemmeno necessario scrivere un'introduzione o prefazione o presentazione, comunque la si voglia chiamare, ma si ritiene che un minimo di spiegazioni o motivazioni per avere scritto una guida tematica sia utile per il lettore.

Se l'utente di questa classe volesse usare una licenza diversa dalla LPPL può farlo, purché non si tratti di una licenza più restrittiva della LPPL stessa. Perciò alla fine dell'introduzione ci sta bene qualche cenno alla licenza. Ci sta bene qualche indicazione in più e forse anche il motivo per avere scelto questa piuttosto che quella licenza. Qualunque licenza si scelga, deve essere una licenza libera altrimenti il G_{IT} non può pubblicare la guida.

Pertanto l'autore di una guida tematica della collana del G_{IT} potrebbe per esempio decidere di usare la licenza della Creative Commons, e indicare le clausole della licenza con le apposite icone. Le possibili combinazioni

di licenze e di mutue interazioni rendono praticamente impossibile creare una macro o un ambiente per specificare quanto la legge impone a seconda delle licenze invocate. Per questo motivo si è scelto di inserire di default la specificazione della LPPL, lasciando all'autore la libertà di specificarne una diversa, ma non più restrittiva.

Spesso la fine dell'introduzione è il posto adatto per inserire il nome dell'autore ed eventualmente anche l'indirizzo di posta elettronica; l'autore esperto non ha nessun problema ad usare una *minipage* di larghezza pari ad una frazione della `\textwidth` con il suo contenuto centrato e con l'indirizzo scritto nella forma `mario dot rossi at server dot network`; naturalmente non è vietato che l'autore scriva il suo indirizzo nella forma `mario.rossi@server.network`, ma è più prudente la prima forma che risulta più difficile da carpire da parte di malware vari.

INFORMAZIONI PER LA PAGINA DEL TITOLO

Potrebbe essere interessante usare il comando `\date` per inserire nella pagina del titolo informazioni relative alla versione del documento. Per non dover ripetere sempre le stesse cose scrivendole varie volte, è opportuno scrivere subito dopo la specificazione della classe da usare con le sue opzioni, la riga che comincia con `\ProvidesFile`; il *<nome del file>* sarà il nome del file sorgente principale della guida tematica; fra le informazioni facoltative (che in questo caso non sono facoltative affatto) *<aaaa/mm/gg>* indica la data di ultima modifica del file; *<v.x.y.z>* indica la versione *x*, subversione *y*, revisione *z*; *<Descrizione>* fornisce con poche parole la descrizione della guida.

Successivamente nel predisporre le informazioni che il comando `\maketitle` userà per comporre il frontespizio si potrà scrivere:

```
\author{<autore o autori>}
\title{<titolo>}
\subtitle{<eventuale sottotitolo>}
\GetFileInfo{<nome del file completo di estensione>}
\date{Versione \fileversion\ del \filedate}
<Eventuale dichiarazione di una licenza diversa dalla LPPL>
\maketitle
```

1.2. USO DELLA CLASSE

ripetendo il *<nome del file completo di estensione>* solo nell'argomento di `\GetFileInfo`, ma senza bisogno di riscrivere i dati ad ogni aggiornamento del file stesso.

SEZIONAMENTI

I comandi di “sezionamento” `\frontmatter`, `\mainmatter` e `\backmatter` indicano sezioni ben definite, ma tocca all'autore indicarli esplicitamente nei posti opportuni.

POSTFAZIONE

La postfazione, qualunque titolo essa abbia, è evidentemente facoltativa e non è assolutamente necessario scriverne una soltanto perché c'è la possibilità di farlo; d'altronde, se si è abituati a scrivere una breve sezione introduttiva alla lettura della bibliografia, non ci si dimentichi che la bibliografia consente di comporre un breve testo prima dell'elenco dei riferimenti, come spiegato qui sotto.

TESTO INTRODUTTIVO ALLA BIBLIOGRAFIA

Per la bibliografia si sfruttano i comandi interni di `|memoir|` per ottenere quello che si desidera e nello stesso tempo per consentire di mettere un prologo prima dell'elenco bibliografico. La documentazione di *memoir* dice che bisogna ridefinire il comando interno `\bibsection`.

```
\renewcommand{\bibsection}{\chapter{\bibname}\prebibhook}
```

Inoltre, per comporre un breve testo da inserire fra il titolo della bibliografia e l'elenco bibliografico vero e proprio, bisogna passarlo come argomento al comando `\prebibhook`; siccome questo comando è predefinito per contenere una stringa nulla, il breve testo deve venire assegnato a `\prebibhook` ridefinendolo mediante `\renewcommand`.

Bisogna specificare un punto che non è affatto marginale: nelle precedenti versioni della classe *guidatematica*, veniva caricata di default il pacchetto *natbib* specificando lo stile con `\bibliographysyle{guidatematica}`, uno stile confezionato apposta per questa classe, in modo da avere i riferimenti identificati solo dai nomi degli autori e l'anno di pubblicazione oltre alle altre informazioni obbligatorie e facoltative; le citazioni sono del tipo “autore-anno”. Questo stile è specifico della classe, e sarebbe opportuno non deviare da queste impostazioni. Ma ci sono autori di guide tematiche

CAPITOLO 1. GUIDA RAPIDA ALLA CLASSE

che hanno bisogno di usare gli stili di *biblatex* e del programma biber per estrarre i riferimenti da un database bibliografico e confezionare il file .bbl con il quale comporre la bibliografia. Se *natbib* e *guidatematica.bst* vengono precaricati, si genera un errore se l'autore specifica di voler usare *biblatex*. Per rimuovere questo vincolo, la nuova versione di questa classe non carica più *natbib* con il suo stile speciale, ma resta a carico dell'utente il compito di caricare i pacchetti che preferisce con gli stili che preferisce. È comunque importante che non si allontani troppo dallo stile preconfezionato per queste guide tematiche.

UNO O PIÙ INDICI ANALITICI

L'indice analitico non è quasi mai necessario; più indici analitici ancor meno. Ma l'autore che voglia comporre uno o più indici analitici è reindirizzato alla documentazione del pacchetto *imakeidx* che consente di farlo nella maniera più comoda finora sviluppata per la loro composizione. Tenga presente che se vuole comporre più indici analitici dovrebbe ridefinire un bel numero di comandi interni della classe, indicati esplicitamente nella documentazione della classe *guidatematica.pdf*.

SINOSI DEI COMANDI E DEGLI AMBIENTI SPECIFICI DELLA CLASSE

`\meta` per definire un argomento `\meta{argomento} → ⟨argomento⟩`
`\marg` per definire un argomento obbligatorio `\marg{argomento} → {⟨argomento⟩}`
`\oarg` per definire un argomento facoltativo `\oarg{argomento} → [⟨argomento⟩]`
`\garg` per definire le coordinate per un ambiente grafico `\garg{x,y} → (⟨x⟩,⟨y⟩)`
`\comando` per comporre un comando senza eseguirlo
`\comando{\comando} → \comando`
`\cs` per comporre un comando usandone solo il nome, inviandone eventualmente le informazioni necessarie all'indice analitico
`\cs{comando} → \comando`
`\Bambiente` per comporre l'apertura di un ambiente specifico
`\Bambiente{ambiente} → \begin{ambiente}`
`\Eambiente` per comporre la chiusura di un ambiente specifico
`\Eambiente{ambiente} → \end{ambiente}`

1.2. USO DELLA CLASSE

- `\Sambiente` per comporre la sintassi dell'apertura di un ambiente con un solo argomento obbligatorio `\Sambiente{ambiente}{N.arg.}` → `\begin{<ambiente>}[<N.arg.>] ... \end{<ambiente>}`
- `\Dambiente` per comporre la sintassi dell'apertura di un ambiente con più argomenti di cui il primo è opzionale e il suo valore di `<default>` viene specificato `\Dambiente{ambiente}{N.arg.}{default}` → `\begin{<ambiente>}[<N.arg.>][<default>] ... \end{<ambiente>}`
- `\Arg` per comporre un argomento specifico racchiuso fra graffe `\Arg{argomento}` → `{argomento}`; il comando `\Marg` ne è un sinonimo.
- `\GetFileInfo` Per ricavare le informazioni relative ad un file provvisto al suo inizio del comando `\Provides...: \GetFileInfo{book.cls}`; non produce risultati compositivi, ma assegna nuovi valori alle variabili `\filename`, `\fileversion` e `\filedate`. Oltre al comando `\ProvidesFile` sono disponibili anche i comandi `\ProvidesClass` e `\ProvidesPackage` per fornire le informazioni relative al loro contenuto, alla loro versione e alla data dell'ultima revisione. Sono comandi disponibili già dal nucleo di \LaTeX ; sono poco conosciuti, ma il loro uso dovrebbe essere "obbligatorio".
- `\setcopymark` scrive in verticale nel margine esterno quello che si desidera: `\setcopymark{Composizione del \today}` → vedi la pagina [29](#)
- `\licenza` è il comando da usare se si desidera specificare una licenza diversa dalla LPPL. La sintassi è semplicemente `\licenza{<dichiarazione>}`, dove `<dichiarazione>` può contenere diversi capoversi e materiale grafico che però deve restare tutto nella stessa pagina dove compare l'invito ai lettori ad iscriversi al [GJt](#). Piuttosto, se la `<dichiarazione>` dovesse davvero essere lunga, si specifichi semplicemente `\licenza{\relax}` (per evitare di inserire la dichiarazione relativa alla LPPL) e si componga una nuova pagina legale tutta a disposizione della 'dichiarazione' che, ovviamente, ora non sarà più necessario racchiudere nell'argomento di `\licenza`.
- `\maketitle` comando ridefinito in modo che componga la pagina del titolo in modo uniforme fra tutte le guide tematiche della collezione e provveda al retrofrontespizio con l'invito a iscriversi al [GJt](#).
- `\subtitle` ulteriore comando per specificare un sottotitolo da comporre nella pagina del titolo: `\subtitle{<Sottotitolo della guida>}` →

Sottotitolo della guida

- `\makeindex` comando modificato per l'uso del pacchetto *imakeidx*
- `\showindexmarks` mostra le parole indicizzate nella parte superiore del margine esterno della pagina; talvolta fallisce se le corrispondenti stringhe sono troppo lunghe
- `\hideindexmarks` nasconde l'indicazione delle parole indicizzate
- `\texttm` comando testuale per comporre col font teletype monospaziato anche quando fosse in vigore quello proporzionale
- `\texttv` comando testuale per comporre col font teletype proporzionale anche quando fosse in vigore quello monospaziato
- `\tvfamily` dichiarazione corrispondente al comando `\texttv \textl` comando per comporre con i font typewriter con le cifre “maiuscole” anche quando fosse in vigore lo stile `OldStyle` per le cifre `\textl{123}` → 123 a confronto con `123` → 123
- `\prebibhook` comando per inserire un (breve) testo fra il titolo della bibliografia e l'elenco bibliografico
- `\ambstyle` comando per comporre il nome degli ambienti con un certo stile di scrittura `\ambstyle{ambiente}` → *ambiente*
- `\classtyle` comando per comporre il nome delle classi con un certo stile di scrittura `\classtyle{classe}` → *classe*
- `\filestyle` comando per comporre il nome dei file con un certo stile di scrittura `\filestyle{file}` → *file*
- `\packstyle` comando per comporre il nome dei pacchetti con un certo stile di scrittura `\packstyle{pacchetto}` → *pacchetto*
- `\progstyle` comando per comporre il nome dei programmi con un certo stile di scrittura `\progstyle{programma}` → *programma*
- `\opzstyle` comando per comporre il nome delle opzioni con un certo stile di scrittura `\opzstyle{opzione}` → *opzione*
- `\contastyle` comando per comporre il nome dei contatori con un certo stile di scrittura `\contastyle{contatore}` → *contatore*
- `\stilestyle` comando per comporre il nome degli stili di pagina con un certo stile di scrittura `\stilestyle{stile}` → *stile*
- `\numeristyle` comando per comporre il nome degli stili di numerazione con un certo stile di scrittura `\numeristyle{arabic}` → *arabic*
- `\umisurastyle` comando per comporre il nome delle unità di misura con un certo stile di scrittura `\umisurastyle{mm}` → *mm*
- `\chiavestyle` comando per comporre il nome delle chiavi nella sintassi

1.2. USO DELLA CLASSE

- $\langle chiave \rangle = \langle valore \rangle$ con un certo stile di scrittura
`\chiavestyle{angle}` \rightarrow angle
- `\descriettorestyle` comando per comporre il codice dei descrittori delle colonne con un certo stile di scrittura `\descriettorestyle{c}` \rightarrow c
- `\posizionestyle` comando per comporre il codice dei posizionamenti degli oggetti mobili con un certo stile di scrittura
`\posizionestyle{t}` \rightarrow t
- `\allineamentostyle` comando per comporre il codice degli allineamenti degli ambineti *tabular*, *array* e simili con un certo stile di scrittura
`\allineamentostyle{b}` \rightarrow b
- `\opz` compone una opzione nello stile `\opzstyle` e manda le informazioni all'indice analitico sotto la voce "opzione"
- `\conta` compone il nome di un contatore nello stile `\contastyle` e manda le informazioni all'indice analitico sotto la voce "contatore"
- `\stile` compone il nome di uno stile di pagina nello stile `\stilestyle` e manda le informazioni all'indice analitico sotto la voce "stile della pagina"
- `\numeri` compone il nome di uno stile di numerazione nello stile `\numeristyle` e manda le informazioni all'indice analitico sotto la voce "numerazione"
- `\misura` compone il nome di una unità di misura nello stile `\umisurastyle` e manda le informazioni all'indice analitico sotto la voce "unità di misura"
- `\chiave` compone il nome di una chiave nello stile `\chiavestyle` e manda le informazioni all'indice analitico sotto la voce "chiave"
- `\descriettore` compone il codice di un descrittore di colonna nello stile `\descriettorestyle` e manda le informazioni all'indice analitico sotto la voce "descrittore di colonna".
- `\posizione` compone il codice di posizionamento di un oggetto mobile nello stile `\posizionestyle` e manda le informazioni all'indice analitico sotto la voce "posizione degli oggetti flottanti"
- `\allineamento` compone il codice di allineamento di una tabella o di una matrice nello stile `\allineamentostyle` e manda le informazioni all'indice analitico sotto la voce "codice di allineamento"
- `\amb` compone il nome di un ambiente nello stile `\ambstyle` e manda le informazioni all'indice analitico sotto la voce "ambiente"

CAPITOLO 1. GUIDA RAPIDA ALLA CLASSE

- `\class` compone il nome di una classe nello stile `\classtyle` e manda le informazioni all'indice analitico sotto la voce "classe"
- `\file` compone il nome di un file (con la sua estensione) oppure la sola estensione di un file composti nello stile `\filestyle` e manda le informazioni all'indice analitico sotto la voce "file"
- `\pack` compone il nome di un pacchetto nello stile `\packstyle` e manda le informazioni all'indice analitico sotto la voce "pacchetto"
- `\prog` compone il nome di un programma nello stile `\progstyle` e manda le informazioni all'indice analitico sotto la voce "programma"
- description* si riferisce al solito ambiente di \LaTeX salvo che l'etichetta della voce da descrivere è composta in maiuscoletto
- blockdescription* si riferisce ad un ambiente simile a *description* privo però del rientro del margine sinistro; l'etichetta è composta in maiuscoletto
- plaindescription* si riferisce ad un ambiente simile a *description* con il rientro del margine sinistro; l'etichetta è composta in font normale, non in neretto come nell'ambiente descrittivo di \LaTeX
- compactitemize* si riferisce ad una lista semplice (puntata) priva di spazi di separazione fra una voce e la successiva
- compactenumerate* si riferisce ad una lista enumerativa priva di spazi separazione fra una voce e l'altra
- compactdescription* si riferisce ad una lista descrittiva priva di spazi di separazione fra una voce e l'altra; l'etichetta è composta in maiuscoletto
- sintassi* Permette di scrivere la sintassi di uno o più comandi dentro un riquadro
- pdfxmetadata* Permette di inserire i metadati specifici di una data guida i metadati necessari nel caso che si desideri creare un guida archiviabile secondo le norme ISO.
- guidatematica.bst* file di stile per la bibliografia da gestire con *natbib* e che ha i riferimenti nello stile autore-anno
- guidatematica.ist* file di stile per gli indici analitici con i gruppi di voci che iniziano con la stessa lettera separati dal titolo formato dalla lettera iniziale
- guidatematicafiles.ist* analogo ma per un indice analitico tematico relativo solo ai file (vedi pacchetto *imakeidx*)

1.2. USO DELLA CLASSE

Si noti che se non si compone nessun indice analitico i comandi del tipo `\ambstyle` e `\amb`, e le altre coppie analoghe, si comportano nello stesso modo.

I comandi `\showindexmarks` e `\hideindexmarks` talvolta non fanno quello che ci si aspetta; se l'informazione (in corpo `\footnotesize`) è troppo larga per stare completamente nel margine esterno, l'indicazione viene omessa, anche se ce n'è menzione nel file `.log`.

Per maggiori dettagli si vedano i paragrafi specifici che si riferiscono a questi comandi.

La classe si usa nel solito modo dichiarandone il nome nell'argomento del comando iniziale `\documentclass`. Non sono preimpostate opzioni di nessun genere e quindi si possono specificare le opzioni che si desiderano fra tutte quelle che la classe *memoir*.

Tuttavia bisogna fare alcune semplici considerazioni:

- Le guide tematiche verosimilmente vengono lette a schermo; la classe specifica un disegno della pagina simmetrico, per cui la lettura a schermo evita al lettore di vedere lo specchio di stampa spostarsi a destra o a sinistra a seconda della parità del numero di pagina. Non è quindi necessario specificare l'opzione *oneside*, come si fa spesso per comporre documenti da leggere a schermo.
- La lettura a schermo è più comoda se il supporto virtuale della pagina, la "carta virtuale", è di formato B5 invece che A4; l'opzione predefinita per il formato della pagina è *b5paper*; tuttavia l'autore può specificare espressamente una delle opzioni *a4paper*, *a5paper*, *pad* per poter comporre su una carta virtuale, lo schermo di un grande dispositivo *hand held*, di 120 mm per 160 mm, oppure *tablet* per poter avere come carta virtuale lo schermo di un tablet di media grandezza con lo schermo da 90 mm per 120 mm; si consiglia di comporre la guida tematica con l'opzione *b5paper* oppure con l'omissione di qualunque opzione relativa al formato della carta.
- Comunque si ritiene che sia meglio specificare un font normale di 11 punti, perché rende la lettura a schermo più facile; tuttavia per l'opzione *tablet* è forse più adatto un corpo più piccolo, per esempio specificando alla classe l'opzione *9pt*.

Per i motivi di immagine esposti nell'introduzione, è opportuno che i singoli utenti apportino il minor numero di personalizzazioni a quanto

2.1. PACCHETTI PRECARICATI

la classe può produrre da sola; le opzioni descritte sopra sono però tutte accettabili.

Naturalmente l'autore è autorizzato ad usare i pacchetti che gli sono funzionali alla composizione della sua specifica guida tematica; viene in mente, per esempio, il pacchetto *tclosorbox* che Roberto Giacomelli ha usato nella sua *Guida tematica alla riga di comando*. Viene in mente il pacchetto *tkzexample* per descrivere le proprietà grafiche del pacchetto *TikZ* o della classe *beamer* su cui Claudio Fiandrino ha scritto la sua guida tematica *Introduzione alla personalizzazione di Beamer*. Siccome questi pacchetti non sarebbero utili per molte altre guide tematiche, è opportuno che gli autori se li carichino da soli nel preambolo.

Oltre a quanto esposto sopra, si può pensare a qualcosa riguardante i font; per impostazione predefinita vengono usati i font Latin Modern estesi, invocati mediante il pacchetto *cfr-lm* quando si compone con pdfLaTeX. Questo pacchetto introduce molti comandi di personalizzazione, tutti usabili anche con questa classe, che permettono, per esempio, di usare i font teletype proporzionali oppure a spaziatura fissa (predefiniti); di usare le cifre minuscole, oppure maiuscole oppure a spaziatura fissa, specialmente per il corretto incolonnamento nelle tabelle. Ecco, queste sono tutte personalizzazioni “consentite”. Cambiare font, benché tecnicamente possibile, è invece fortemente sconsigliato. I font Latin Modern OpenType sono di default con XeLaTeX e LuaLaTeX.

Il layout della pagina e i margini ai quattro lati dello specchio di stampa non sono modificabili; cioè lo sarebbero, ma si raccomanda agli utenti di non farlo; si snaturerebbe l'aspetto grafico della guida, e si romperebbe quell'uniformità grafica richiesta a tutte le guide tematiche del *gJr*.

La classe carica un certo numero di pacchetti di configurazione e di estensione; se ne usino le funzionalità a piacimento, ma non si installino altri pacchetti, talvolta incompatibili, se sono tali da alterare l'aspetto grafico della classe.

2.1 PACCHETTI PRECARICATI

Alcuni pacchetti precaricati sono funzionali alla classe stessa, altri sono o potrebbero essere utili per l'autore:

iftex è funzionale per la classe, visto che si può usare come programma di compilazione sia pdfLaTeX, sia XeLaTeX sia LuaLaTeX. Per esem-

pio, alcuni dei pacchetti che seguono sono funzionali solo se si usa pdfLaTeX, quindi essi sono caricati solo se si usa pdfLaTeX.

fancyvrb serve per estendere le funzionalità necessarie per creare materiale verbatim; definisce comandi come `\Verb` e ambienti come *Verbatim* che estendono i corrispondenti ambienti del nucleo di L^AT_EX, accettando anche alcune opzioni. Inoltre definisce l'ambiente *VerbatimOut* e il comando `\VerbatimInput` per scrivere in un file esterno del materiale verbatim, e per leggere da un file esterno del materiale da comporre in modo verbatim. Il pacchetto *fancyvrb* è compatibile con la codifica UTF-8, mentre il pacchetto *listing* consente di comporre listati molto più *fancy* ma è incompatibile con la codifica multibyte di UTF-8 e di Unicode in generale.

graphicx serve per inserire e manipolare immagini; disponendo di una distribuzione del sistema T_EX (T_EX Live o MiKTeX molto recente e aggiornata, oltre che completa – non una distribuzione basic, tanto per intenderci) sia pdfLaTeX, sia XeLaTeX sia LuaLaTeX possono inglobare lo stesso tipo di file grafici, cioè quelli con estensione .pdf, .jpg, .png, .eps, .mps. In una prima versione di questa classe si era caricato il pacchetto *guit*, che a sua volta caricava *graphicx*, ma poi si è scoperto che il disegno del logo G_UI_T era incompatibile con X_YL^AT_EX; perciò si è preferito *non* caricare questo pacchetto (e l'autore è avvisato di non caricarlo mai, se vuole usare XeLaTeX, finché non sarà aggiornato per essere compatibile con X_YL^AT_EX). Al suo posto si è generata una immagine abbastanza grande del logo, e si è creata una macro che inserisca il logo nelle giuste proporzioni con il corpo del font in uso; cambiando il corpo dei font cambia corrispondentemente anche l'immagine riprodotta del logo G_UI_T.

metalogo serve per poter descrivere i loghi che vengono generalmente usati per indicare certi compilatori o certe indicazioni particolarmente frequenti quando si scrive di L^AT_EX. Per esempio consente di usare il comando `\XeLaTeX` per scrivere correttamente X_YL^AT_EX con la 'E' speculare.

pict2e nella versione aggiornata dopo il primo di giugno 2009; è funzionale alla classe, ma potrebbe venire usato anche dall'autore.

microtype serve per usare pienamente la microtipografia se si usa pdfLaTeX, sia LuaLaTeX con la protrusione e l'espansione dei font; con XeLaTeX funziona la protrusione ma non ancora l'espansione dei font.

2.1. PACCHETTI PRECARICATI

etoolbox è funzionale alla classe, ma non è escluso che possa essere utile anche all'autore.

xcolor serve per usare i colori e per sfruttare la sua sintassi per definire i colori “misti”; non è specificata nessuna opzione per i colori. L'autore è pregato di non caricare il pacchetto *color* che vanifica in parte le funzionalità di *xcolor* e può produrre incompatibilità.

tikz viene caricato senza opzioni; è funzionale per la classe per la quale si carica anche la sua estensione *decorations*; l'autore non ha quindi bisogno di bisogno di caricare esplicitamente *tikz*, ma se lo fa, lo carichi senza opzioni, per evitare errori di *option clash* ma lo configuri usando le sue estensioni specifiche mediante il comando `\usetikzlibrary` e/o i suoi comandi di setup.

natbib era precaricato dalla classe nelle sue precedenti versioni; ora non è lasciato all'autore insieme al file di stile bibliografico *guidatematica.bst* e se ne raccomanda l'uso. Esso era funzionale alla compilazione della bibliografia che usa uno stile “autore-anno” e si rifà ad una variante dello stile bibliografico *arstexnica.bst* e si chiama *guidatematica.bst*; viene distribuito insieme a questa classe. Ma questo pacchetto precaricato impedisce all'autore di servirsi di *biblatex*. Per rimuovere questo ostacolo, *natbib* non viene più precaricato, e si lascia all'autore il suo caricamento. Piccole personalizzazioni possono essere attuate re-impostando alcune opzioni di *natbib* stesso attraverso i suoi propri comandi descritti nella sua documentazione. Ora che l'ostacolo è rimosso, l'autore deve caricare *natbib* con lo stile raccomandato, oppure *biblatex* con uno stile non troppo difforme, da quello predisposto per questa classe.

multicol può servire all'autore, ma è funzionale alla composizione dell'indice analitico.

imakeidx serve per comporre uno o più indici analitici in modo sincrono con la composizione del documento. Esso era inizialmente incompatibile con la classe *memoir* che aveva un suo meccanismo per comporre uno o più indici analitici; dalla versione 1.1 in poi *imakeidx* consente di comporre gli indici analitici anche con *memoir* purché si usino solo le impostazioni e i comandi di *imakeidx*. Dalla versione 1.2e *imakeidx* è compatibile anche con le funzionalità di *showidx* originali, sia con quelle simulate dalla classe *memoir*. Avendo la classe *memoir* come base, non è necessario caricare mai il pacchetto

to *showidx*, ma per mostrare le voci indicizzate nel margine basta specificare `\showindexmarks` dopo `\begin{document}` per attivare questa funzionalità; `\hideindexmarks` la disattiva; questa funzionalità è inizialmente disattivata ma, appunto, può essere attivata e disattivata dall'autore a suo piacimento.

inputenc è caricato dalla classe con l'opzione di codifica *utf8* se e solo se si sta componendo con pdfLaTeX. Con XeLaTeX e LuaLaTeX si deve usare solo la codifica *utf8* senza specificare il pacchetto *inputenc*. Avendo caricato questo pacchetto con questa opzione, si pensa di avere agevolato l'autore, anche se è possibile che, per comporre con questa classe il file di una guida tematica già scritto in precedenza, egli debba provvedere a cambiarne la codifica.

fontenc è caricato se e solo se si sta componendo con pdfLaTeX; l'opzione prescelta è *T1*; con XeLaTeX e LuaLaTeX si usano solitamente font OpenType in codifica UNICODE, quindi la scelta dei font con cui comporre il documento deve essere fatta dall'autore. Questa classe carica di default i font Latin Modern Type 1 quando si compone con pdfLaTeX e Latin Modern OpenType quando si compone con XeLaTeX o LuaLaTeX.

babel viene caricato solo con l'opzione *italian* e solo se si sta componendo con pdfLaTeX; se si vogliono usare altre lingue oltre all'italiano se ne specifichino le opzioni fra quelle della classe. Se invece si compone con XeLaTeX o LuaLaTeX, il pacchetto *polyglossia* è già precaricato ed è già specificato che la lingua principale è l'italiano, ma si possono dichiarare direttamente altre lingue (secondarie/alternative) secondo le modalità proprie di *polyglossia* con comandi da inserire nel preambolo.

cfr-lm quando si compone con pdfLaTeX serve per usare i font vettoriali Latin Modern in modo avanzato, sfruttandone le molte possibilità di uso che non sono accessibili con il semplice pacchetto *lmodern*. Il lettore si può riferire alla documentazione del pacchetto *cfr-lm* per avere una panoramica completa dell'uso avanzato che si può fare con questo pacchetto. Quando una guida viene composta con pdfLaTeX viene usato proprio il pacchetto *cfr-lm*.

fontspec se invece si compone con XeLaTeX o con LuaLaTeX, viene caricato il pacchetto *fontspec* senza opzioni; si sono definiti alcuni font e alcuni comandi per far sì che usare pdfLaTeX, XeLaTeX, o LuaLaTeX

2.2. NUOVI COMANDI

non producano risultati sostanzialmente differenti. Si veda però il capitolo 4.

polyglossia viene richiamato se si compone con XeLaTeX o con LuaLaTeX e viene impostato l'italiano come lingua principale; se l'autore desidera usare altre lingue, basta che nel suo preambolo specifichi `\setotherlanguages` elencando le altre lingue che desidera usare con la solita lista di lingue separate da virgole.

fancyvrb può essere utile all'autore per comporre ambienti verbatim in modo diverso da quello che si può ottenere con l'ambiente standard del nucleo di L^AT_EX. Se ne legga la documentazione prima di farne uso.

afterpage serve per trattenere l'esecuzione di una sezione di codice fino alla fine della pagina corrente; molto utile in certe circostanze, ma qui finalizzato soltanto all'uso di questa classe.

enumitem serve per personalizzare le liste, crearne di nuove, specificare ogni dettaglio di composizione delle etichette, eccetera; con le funzionalità di questo pacchetto si sono ridefiniti numerosi ambienti utili non solo per comporre questo documento, ma anche per l'autore in generale che usa la classe *guidatematica* per descrivere un tema particolare inerente il mondo di T_EX ed abbia bisogno di nuovi tipi di liste. La consultazione della documentazione di *enumitem* diventa quindi utile e necessaria per usare i comandi di personalizzazione in modo corretto ed efficiente.

listings è il pacchetto principe per comporre listati di comandi di vari programmi; in questa classe esso è già configurato per usare la sintassi valida per il linguaggio T_EX nel suo 'dialetto' L^AT_EX. Possono venire caricati altri dialetti come si possono caricare altri linguaggi. Si possono anche estendere le liste di parole chiave secondo le modalità descritte nella documentazione del pacchetto *listings*.

2.2 NUOVI COMANDI

La classe *guidatematica* mette a disposizione dell'autore un certo numero di comandi per rendere più agevole la descrizione della sintassi di certi comandi e di certi ambienti e, eventualmente, di inviare agli indici analitici i comandi, i nomi dei pacchetti, dei programmi, degli ambienti, eccetera.

`\meta` serve per scrivere il descrittore di un argomento in corsivo fra parentesi acute; per esempio: $\langle numero \rangle$.

`\marg` serve per racchiudere il descrittore di un argomento fra parentesi graffe composte in carattere teletype; per esempio: $\{\langle numero \rangle\}$.

`\oarg` serve per racchiudere il descrittore di un argomento fra parentesi quadre composte in carattere teletype; per esempio: $[\langle numero \rangle]$.

`\garg` richiede due descrittori di argomenti separati da una virgola e serve per comporli fra parentesi tonde, sempre separati da una virgola; virgola e parentesi tonde sono composte in carattere teletype; per esempio: `\garg{x1,y1}` produce $(\langle x1 \rangle, \langle y1 \rangle)$.

`\comando` serve per scrivere un comando verbatim, senza interpretarlo e senza svilupparlo; per esempio: `\textup`.

`\cs` è simile a `\comando` ma richiede il nome del comando senza il backslash iniziale ma ne stampa il nome preceduto dal suo backslash; per esempio: `\normalfont`. Se vengono composti gli indici analitici, invia il comando all'indice analitico generale completo di backslash ma indicizzato senza backslash.

`\Sambiente` serve per descrivere la sintassi di un ambiente, il cui nome ne costituisce il primo argomento; il secondo argomento rappresenta il numero di argomenti che il comando di apertura può ricevere; per esempio:

```
\begin{\langle ambiente \rangle}[\langle N-arg \rangle] ... \end{\langle ambiente \rangle}
```

Per ambienti che richiedono sia il numero di argomenti, sia il valore di default del primo argomento, se questo è opzionale, bisogna procedere con il prossimo comando.

`\Dambiente` si comporta come il comando precedente ma compone due argomenti facoltativi, di cui il secondo rappresenta il valore di default del primo argomento che si può passare al comando di apertura; per esempio:

```
\begin{\langle ambiente \rangle}[\langle N-arg \rangle][\langle default \rangle] ... \end{\langle ambiente \rangle}
```

`\Bambiente` serve per comporre il comando di apertura di un ambiente specifico in carattere teletype; per esempio:

```
\begin{equation}.
```

`\Eambiente` similmente serve per scrivere il comando di chiusura di un ambiente specifico; per esempio: `\end{equation}`.

2.2. NUOVI COMANDI

`\Arg` e `\Marg` sono equivalenti; servono per comporre in carattere teletype un argomento specifico racchiuso fra le sue parentesi graffe; per esempio: `{Titolo di questo documento}`.

`\Oarg` simile a `\Marg` ma si riferisce agli argomenti facoltativi: `[twoside]`.

`\GetFileInfo` richiede un argomento costituito dal nome di un file completo di estensione; se il file è dotato del comando `\ProvidesFile`, oppure `\ProvidesClass` oppure `\ProvidesPackage`, allora questo comando estrae le informazioni che attribuisce alle macro `\filename`, `\filedate` e `\fileversion` così da poterle usare per altri scopi, per esempio con il prossimo comando. Il file di cui estrarre le informazioni deve essere già stato caricato; per esempio: il file principale con cui è composto questa guida si chiama `guidatematica-doc.tex` e questo nome passato come argomento a `\GetFileInfo` produce le tre informazioni: `\filename = guidatematica-doc.tex`; `\filedate = 2020-04-29`; e `\fileversion = v.2.0.01`. Ma il comando `\documentclass` carica la classe `guidatematica`, perciò il comando `\GetFileInfo{guidatematica.cls}` produce le tre informazioni: `\filename = guidatematica.cls`; `\filedate = 2020-04-29`; e `\fileversion = v.2.1.00`.

`\setcopymark` compone il suo argomento ruotato di 90° nella parte bassa del margine esterno; serve per indicare il tipo di bozza che si è appena composta, oppure qualunque altra informazione che si desidera ripetere ad ogni pagina; di per sé il comando non prevede nessun testo predefinito; quindi sta all'autore usare un argomento adatto per avere le informazioni che gli servono; qui, per esempio, dopo aver eseguito il precedente comando `\GetFileInfo` in questa maniera:

```
\GetFileInfo{guidatematica-doc.tex}
```

si usa (solo per questa pagina) il seguente argomento:

```
\setcopymark{%  
  \textcolor{blue!50!white}{%  
    \filestyle{\filename},  
    \fileversion,  
    del \filedate}}
```

È ovvio, però, che questo genere di informazioni dovrebbe apparire su tutte le pagine di una bozza e dovrebbe venire cancellato solo per

la versione definitiva.

`\licenza` permette di comporre una diversa dichiarazione a cui la guida tematica può essere soggetta. Per impostazione predefinita la licenza a cui dovrebbero essere sottoposte tutte le guide tematiche è la LPPL (L^AT_EX Project Public Licence), quella a cui sono soggetti quasi tutti i programmi e i file del sistema T_EX; tuttavia in certe circostanze potrebbe essere necessario specificare una licenza diversa. Per esempio i file che contengono i pattern di sillabazione sono soggetti ad altre licenze, perché vengono usati anche da altri software più o meno liberi che hanno adottato l’approccio del sistema T_EX per gestire la sillabazione. Tolti questi casi particolari, gli altri software del sistema T_EX sono generalmente soggetti alla LPPL. Nel caso delle guide tematiche del G_LT potrebbero essere coinvolti anche programmi esterni a quelli del sistema T_EX; in qualche guida tematica potrebbe essere quindi necessario specificare una licenza che permetta l’uso di quel software. Il comando `\licenza` permette di definire una dichiarazione di licenza che va a sostituire, nel retro del frontespizio, la dichiarazione relativa alla LPPS. La sintassi è semplicissima:

<code>\licenza{<dichiarazione relativa alla nuova licenza>}</code>
--

dove l’argomento potrebbe contenere anche alcuni (breve) capoversi e, per esempio, una serie di piccole icone come quelle predisposte dalla Creative Commons.

Attenzione: se l’autore di una guida tematica è “tenuto” ad usare una licenza diversa dalla LPPL, deve usare questo comando `\licenza`, ma lo deve usare *prima* di generare la pagina del titolo mediante il comando `\maketitle`. Se invece l’autore deve comporre un’intera pagina per queste informazioni legali, allora deve prima inibire la produzione della dichiarazione relativa alla LPPL; la via più semplice e intuitiva consiste nello specificare `\licenza{}` perché in questo modo l’argomento di `\licenza` è vuoto, e il test per verificarne il contenuto evita di scrivere alcunché in calce al retrofrontespizio.

I comandi seguenti definiscono lo stile di scrittura di determinate entità e per scriverle in quello stile; non mettono niente, eventualmente, negli indici analitici. Sarebbe opportuno non ridefinire questi comandi, perché sarebbe desiderabile che in tutte le guide tematiche i nomi delle classi

2.2. NUOVI COMANDI

venissero sempre composti in un dato carattere, i nomi dei pacchetti in un altro dato carattere, eccetera, altrimenti i lettori ne resterebbero confusi.

`\ambstyle` serve per comporre il nome di un ambiente in carattere lineare inclinato; per esempio: *description*.

`\classstyle` serve per comporre il nome di una classe in carattere teletype proporzionale inclinato; per esempio: *guidatematica*.

`\filestyle` serve per comporre il nome di un file in carattere teletype monospaziato diritto; per esempio: `tlenc.def`.

`\packstyle` serve per comporre il nome di un pacchetto in carattere teletype proporzionale corsivo; per esempio: *microtype*.

`\progstyle` serve per comporre il nome di un programma in carattere lineare diritto; per esempio: `pdfLaTeX`.

`\opzstyle` serve per comporre il nome di un'opzione in carattere graziato corsivo; per esempio: *landscape*.

`\contastyle` serve per comporre il nome di un contatore in carattere teletype monospaziato diritto; per esempio: `page`.

`\stylestyle` serve per comporre il nome dello stile di pagina in carattere teletype monospaziato diritto; per esempio: `headings`.

`\numeristyle` serve per comporre il tipo di numerazione in carattere teletype monospaziato diritto; per esempio: `arabic`.

`\umisurastyle` serve per comporre il nome di una unità di misura del sistema $\mathrm{T\!E\!X}$ in carattere teletype monospaziato diritto; per esempio: `pt`.

`\chiavestyle` serve per comporre le varie chiavi che $\mathrm{L\!A\!T\!E\!X}$ usa nelle scritture del tipo “ $\langle chiave \rangle = \langle valore \rangle$ ” in carattere teletype monospaziato diritto; per esempio: `angle`.

`\descrittorestyle` serve per comporre i descrittori delle colonne delle tabelle e delle matrici in carattere teletype monospaziato diritto; per esempio: `p`.

`\posizionestyle` serve per comporre i simboli di posizione degli oggetti flottanti in carattere teletype monospaziato diritto; per esempio `h`.

`\allineamentostyle` serve per comporre i simboli dell'allineamento verticale di tabelle, matrici, ambienti *minipage*, e simili in carattere teletype monospaziato diritto; per esempio: `c`.

Va detto che l'uso di questi comandi non è necessario quando non si compongono gli indici analitici per i quali sono funzionali nel senso di

comporre la parola nel testo e nell'indice usando lo stesso stile; potrebbe essere necessario usarli nei comandi di sezionamento quando si vuole evitare che i collegamenti ipertestuali generati da *hypertext* (non precaricato nella classe, ma da richiedere espressamente da parte dell'autore) non abbiano da "singhiozzare" perché certi comandi non sono accettabili nei testi che possono costituire dei bookmark. In questo caso si dovrebbe procedere usando il comando `\texorpdfstring` in questo modo:

```
\subsection{L'uso della chiave %
  \texorpdfstring{\chiavestyle}{\angle}}
```

I comandi seguenti fanno uso dei comandi precedenti e, se è richiesta la composizione degli indici analitici, provvedono a inviare ai debiti file di indicizzazione l'informazione corretta anche quando si tratta di termini da raggruppare sotto una voce principale. Perciò questi comandi sono quasi del tutto equivalenti ai precedenti comandi se *non* è stato specificato il comando `\makeindex` nel preambolo, perché solo questo comando rende attivo ed efficace il comando `\index`.

- `\opz` compone il suo argomento nello stile delle opzioni e lo invia all'indice analitico principale sotto la voce "opzione".
- `\conta` compone il suo argomento nello stile dei contatori e lo invia all'indice analitico principale sotto la voce "contatore".
- `\stile` compone il suo argomento come lo stile delle pagine e lo invia all'indice analitico principale sotto la voce "stile della pagina".
- `\numeri` compone il suo argomento nello stile delle numerazioni e lo invia all'indice analitico principale sotto la voce "numerazione".
- `\umisura` compone il suo argomento nello stile delle unità di misura e lo invia all'indice analitico principale sotto la voce "unità di misura".
- `\chiave` compone il suo argomento nello stile delle chiavi e lo invia all'indice analitico principale sotto la voce "chiave".
- `\descrittore` compone il suo argomento nello stile dei descrittori di colonna e lo invia all'indice analitico principale sotto la voce "descrittore di colonna".
- `\posizione` compone il suo argomento nello stile della posizione degli oggetti flottanti e lo invia all'indice analitico principale sotto la voce "Posizione degli oggetti flottanti".

2.2. NUOVI COMANDI

- `\allineamento` compone il suo argomento nello stile dei codici di allineamento e lo invia all'indice analitico principale sotto la voce "codice di allineamento".
- `\amb` compone il suo argomento nello stile degli ambienti e lo invia all'indice analitico principale sotto la voce "ambiente".
- `\class` compone il suo argomento nello stile delle classi e lo invia all'indice analitico principale sotto la voce "classe".
- `\file` compone il suo argomento nello stile dei file e lo invia all'indice analitico principale sotto la voce "file".
- `\pack` compone il suo argomento nello stile dei pacchetti e lo invia all'indice analitico principale sotto la voce "pacchetto".
- `\prog` compone il suo argomento nello stile dei programmi e lo invia all'indice analitico principale sotto la voce "programma".

2.3 NUOVI AMBIENTI

Grazie al pacchetto *enumitem* si sono fortemente estese le possibilità di personalizzare le liste rispetto a quanto si può fare con i comandi di personalizzazione di *memoir*. I suoi ambienti standard *description* e *block-description*, sono stati modificati per avere la loro etichetta composta in maiuscoletto; inoltre sono state definite altre liste che si descrivono qui di seguito.

Si noti che per le liste descrittive si può cambiare a piacere il modo di scrivere l’oggetto da descrivere, e questo torna molto utile in una classe come *guidatematica*, dove vengono descritti spesso oggetti della stessa “categoria”, che spesso debbono essere composti in uno stile particolare oppure devono venire inviati all’indice analitico. Si veda poco oltre alcuni esempi di questi usi che ricorrono alla comoda sintassi del pacchetto *enumitem*, basata su espressioni del tipo $\langle \textit{chiave} \rangle = \langle \textit{valore} \rangle$.

`DESCRIPTION` non differisce dall’omonimo ambiente del nucleo di \LaTeX e della classe *book*; come è stato ridefinito qui, ha l’oggetto da descrivere composto in maiuscoletto.

`BLOCKDESCRIPTION` è costituito da questa particolare descrizione più compatta orizzontalmente di quella tradizionale di \LaTeX , ma altrettanto aperta verticalmente, con le etichette non sporgenti a sinistra e senza margine di lista rientrato. Anche in questo ambiente l’oggetto da descrivere è composto in maiuscoletto.

`COMPACTITEMIZE` produce la solita lista puntata in modo compatto verticalmente, come si è fatto sempre in questa guida. Infatti, tranne in questo particolare ambiente per comporre questa descrizione dei nuovi ambienti, si può notare che gli spazi verticali sono stati eliminati da tutte le liste, e si è lasciato solamente uno spazio di un quarto di riga prima dell’inizio e dopo la fine di tutte le liste.

`COMPACTENUMERATE` produce una lista numerata verticalmente compatta; differisce da *compactitemize* solo per il modo di scrivere l’etichetta di ogni voce.

`COMPACTDESCRIPTION` produce una lista descrittiva verticalmente compatta. Si è ritenuto che questo ambiente fosse quello più indicato per descrivere

2.3. NUOVI AMBIENTI

i nuovi comandi introdotti in questa classe e per crearne le corrispondenti voci nell'indice analitico. Lo si è fatto nel modo descritto qui di seguito.

PLAINDESCRIPTION differisce dall'ambiente *description* solo perché il modo di scrivere l'etichetta è impostato a `\normalfont`; questo ambiente si presta particolarmente bene per usare stili diversi per le varie etichette degli oggetti da descrivere.

1. Questo ambiente *plainedescription* o l'ambiente *compactdescription* (come d'altronde si può fare con qualunque altro ambiente gestito con il pacchetto *enumitem*) sono particolarmente adatti per i tipi di ambiente che descrivono comandi o altre cose simili di un linguaggio di programmazione, di una classe, di un pacchetto, perché con il comando di apertura è semplice specificare come argomento facoltativo il modo di comporre l'etichetta dell'oggetto da descrivere; per esempio, se si volessero descrivere un certo numero di file, varrebbe la pena descriverli tutti con lo stile per comporre i nomi dei file; basta allora fare così:

```
\begin{[plainedescription}%  
      [format=\ttfamily]
```

usando nella descrizione del formato solo delle dichiarazioni¹; quali dichiarazioni? Quelle usate per definire `\filestyle`. Per la verità la definizione di `\filestyle` contiene una dichiarazione e un comando perché deve venire usata anche nelle specificazioni dell'oggetto da indicizzare, ma non è difficile aprire il file `guidatematica.cls`, cercare la definizione di `\filestyle` e ricavare da lì le dichiarazioni da usare. Ci sono sostanzialmente due modi per sfruttare questi ambienti: il primo consiste nell'usare *plainedescription* e scrivere ogni voce della lista nella forma, per esempio:

¹Nella documentazione di *enumitem* c'è scritto che l'ultimo comando che specifica la chiave `format`, o il suo sinonimo `font`, può accettare un argomento; quindi lascerebbe intendere che l'ultima sequenza di controllo potrebbe essere un comando, invece che una dichiarazione. Non spiega però come fare. Si fa così: basta racchiudere fra graffe il contenuto delle quadre che contengono l'oggetto da descrivere. Questo modo di procedere può essere fonte di dimenticanze (bisogna scrivere `\item[{\langle voce \rangle}]` invece che semplicemente `\item[\langle voce \rangle]`, quindi è facile dimenticare il paio di graffe interne). Per quanto sia possibile seguire questa strada forse è preferibile seguirne un'altra descritta poco oltre.

```
\item[\file{\file}]
```

Il secondo modo consiste nell’usare l’argomento facoltativo del comando di apertura specificando con la sintassi $\langle chiave \rangle = \langle valore \rangle$ le opzioni compositive dell’etichetta dell’oggetto da descrivere.

Il vantaggio di seguire questa strada è che dopo aver specificato il $\langle valore \rangle$ della chiave `format`, in tutta la lista si userà la normale sintassi `\item[\file]` per avere tutti i nomi dei file, che iniziano ogni voce della descrizione, composti tutti con lo stesso carattere in modo omogeneo con qualunque altra istanza dello stesso nome di file in qualunque altro punto del documento.

2. In particolare, se si volessero descrivere delle entità particolari da raggruppare nell’indice analitico sotto una stessa voce si potrebbe invocare questo ambiente con il comando:

```
\begin{compactdescription}[before=%
  {\let\makelabel\indicizzafile}]
```

dopo aver definito in questo modo il comando `\indicizzafile`:

```
\DeclareRobustCommand
  \indicizzafile[1]{\file{#1}}
```

oppure si potrebbe agire direttamente²:

```
\begin{compactdescription}[before={%
  \renewcommand\makelabel[1]{%
    \file{##1}}}]
```

in modo che con l’una o l’altra definizione si potrebbe scrivere:

```
... i file di servizio sono:
\begin{compactdescription}[before=%
  {...}]
\item[fleqn.clo] serve per comporre
  le equazioni ad una distanza fissa
```

²La chiave `before` mette il suo argomento in testa all’elenco di comandi che L^AT_EX, internamente, esegue per impostare una lista. Il doppio “cancellotto” `##` prima del numero dell’argomento è necessario quando si sta definendo una macro con argomento all’interno di un’altra macro.

2.3. NUOVI AMBIENTI

```
dal margine sinistro della pagina
invece che centrate.
\item[leqno.clo] serve per scrivere
il numero di ogni equazione
numerabile a sinistra invece che
a destra.
...
```

Naturalmente l'autore, se vuole, può definire altre varianti dell'ambiente di descrizione sfruttando i comandi del pacchetto *enumitem* predisposti a questo scopo.

Sono stati ridefiniti anche gli ambienti *thebibliography* e *theindex*, in modo che compaiano anche nell'indice generale; l'ambiente *theindex* è reso compatibile con il pacchetto *imakeidx* in modo da poterne sfruttare tutte le funzionalità. Probabilmente non era necessario ridefinirlo, ma non disturba averlo fatto.

Per la bibliografia si usa lo stile predefinito *guidatematica.bst*, del tipo autore-anno e adattato alla lingua italiana; se si vuole usare *biblatex* o un altro stile, è possibile farlo anche con la presenza di questa impostazione. Per le citazioni l'autore può usare tutte le forme del comando *\cite* e delle sue varianti messe a disposizione dal pacchetto *natbib*. Se l'autore decide di usare *biblatex* troverà comodo specificare l'apposita opzione per rendere *biblatex* compatibile con i suddetti comandi di *natbib*.

Molto importante per una guida di questo genere è l'ambiente *sintassi* predisposto per specificare la sintassi dei comandi dentro un ambiente sbandierato con l'allineamento a sinistra; di per sé questo ambiente definisce l'ambiente *flushleft* e una cornice dentro la quale specificare la sintassi che si desidera descrivere; di solito si tratta di una riga soltanto, ma se dovesse richiedere più righe allora l'autore sarebbe facilitato se in apertura dell'ambiente specificasse il comando *\obeylines* che trasforma ogni fine riga in un "a capo" esplicito. Per esempio si può descrivere l'impostazione di un documento con la sintassi seguente:

```

\documentclass[<opzioni>]{<classe>}
<preambolo>
\begin{document}
<corpo del documento>
\end{document}

```

con i comandi:

```

\begin{sintassi}\obeylines
\cs{documentclass}\oarg{opzioni}%
    \marg{classe}
\meta{preambolo}
\Bambiente{document}
\meta{corpo del documento}
\Eambiente{document}
\end{sintassi}

```

Inoltre la classe *memoir* non definisce l'ambiente *titlepage* per una esplicita scelta del suo autore. Nel nostro caso, invece si vuole che la pagina del titolo sia composta in modo uniforme per tutte le guide tematiche del GfT.

Per questo scopo l'autore non deve usare esplicitamente questo ambiente; basta che usi il comando `\maketitle`, che a sua volta imposta l'ambiente. L'autore deve limitarsi a specificare nel preambolo i soliti comandi arricchiti dall'ulteriore comando `\subtitle`, il cui uso è evidentemente facoltativo:

```

\author{<un solo nome di autore o una lista di nomi separati da virgole>}
\title{<Titolo della guida>}
\subtitle{<eventuale sottotitolo>}
\date{<data o altra informazione>}

```

Se `\date` non viene specificato, viene inserita la data del giorno; se viene specificata con un argomento nullo, non viene inserito nulla nella pagina del titolo; altrimenti si può specificare quello che si vuole, purché sia una breve locuzione visto che viene scritta nel piedino. Potrebbe essere utile usare il comando `\GetFileInfo` e inserire in `\date` il nome, la data e la versione contenute nelle rispettive macro.

2.4. LA VIRGOLA INTELLIGENTE

L'intero blocco dei dati nella pagina del titolo viene centrato orizzontalmente nella pagina e il logo tondo del $\mathbb{G}\mathbb{I}\mathbb{T}$ viene inserito automaticamente; si tratta di una variante del “timbro” del $\mathbb{G}\mathbb{I}\mathbb{T}$ predisposta in modo che la scritta attorno al logo centrale non abbia mai le lettere capovolte.

2.4 LA VIRGOLA INTELLIGENTE

Come è noto, i numeri fratti vanno scritti con la virgola nella funzione di separatore decimale in tutte le lingue tranne che in inglese e nelle sue varietà. Certo è possibile usare il pacchetto *siunitx* impostandone le opzioni per usare sempre la virgola e per separare la parte intera e la parte fratta in gruppi di tre cifre collegate da uno spazio fine non separabile (quando le cifre sono più di quattro) usando il suo comando `\num`.

Tuttavia questa classe contiene una definizione della virgola come segno *da usare in matematica* in modo che capisca da solo, dal contesto in cui si trova, se funziona da separatore decimale oppure da segno di interpunzione. Si noti che dalla versione 1.3 del file di descrizione della lingua italiana per *babel*, la definizione della virgola intelligente è già contenuta in quel file; se però si usa *polyglossia*, il corrispondente file di descrizione dell'italiano non contiene nessuna definizione per la virgola intelligente, quindi ci pensa la classe a farlo, anche se per attivare questa funzionalità bisogna specificare il comando `\IntelligentComma`; per disabilitarlo bisogna usare il comando `\NoIntelligentComma` anche quando si usi *babel*.

Il fatto che in italiano la virgola matematica svolga due funzioni diverse non è tanto per quel che riguarda i numeri, quanto per le liste di oggetti matematici che compaiono frequentemente quando si scrive tecnico. Queste vanno dall'elenco delle variabili in una funzione (per esempio: $f(x, y, z)$), ad un elenco di funzioni (per esempio: “le funzioni Gx , $H(x)$, e $H(x)$ sono legate da ...”); da una lista di numeri (per esempio: “per ogni $i = 1, 2, 3, \dots$ ”), agli elementi di un insieme (per esempio: “ $\mathbb{Z}_N = 0, 1, 2, 3, \dots, N$ ”). Gli esempi forniti non rappresentano una casistica completa, ma sono indicativi del problema.

Anche se la filosofia dell'uso del mark-up di \LaTeX è quella di sollevare il compositore dall'occuparsi della forma, ma di concentrarsi sul contenuto, scrivendo espressioni matematiche è sufficientemente impegnativo occuparsi della moltitudine di simboli diversi e, se si riesce a sollevare il compositore dall'occuparsi della doppia funzione della virgola, tanto di guadagnato.

La macro della virgola intelligente, dunque, usa la virgola decimale se e solo se è seguita da una cifra *senza spazi interposti*, e usa la virgola interpuntiva nelle liste. L'unico caso in cui bisogna stare attenti ad inserire uno spazio dopo la virgola (nel file sorgente) è quando si usa una elencazione di numeri, come per esempio in `\forall i=1,2,3,\dots,\infty` che produce correttamente $\forall i = 1, 2, 3, \dots, \infty$; in questo caso, infatti, il contesto non è abbastanza significativo per l'“intelligenza” della virgola al fine di decidere quale tipo di funzione essa svolga fra due cifre, se non ci fosse lo spazio interposto. Tralasciando gli spazi tra la virgola e la cifra che la segue si otterrebbe invece $\forall i = 1, 2, 3, \dots, \infty$, che ha le spaziature evidentemente errate. Se ciò che segue la virgola non è una cifra, la definizione della virgola inserisce una virgola interpuntiva, come succede quando essa è seguita dalle sequenze di controllo `\dots` e `\infty`.

Forse è opportuno spiegare perché qui si usi la locuzione “sequenza di controllo” invece della parola “macro” o di altre espressioni più comuni: una sequenza di controllo è in generale un *token complesso*, cioè un token il cui nome è formato da una o più lettere o simboli ed è preceduto dal carattere di escape, generalmente, ma non obbligatoriamente, rappresentato dal segno `\`; potrebbe andare bene qualunque altro carattere che abbia codice di categoria pari a zero. Una sequenza di controllo può essere: (a) indefinita, se non le è mai stato assegnato un significato; (b) una macro se le è stato assegnato un significato mediante il comando primitivo `\def` o un comando equivalente; (c) qualcosa che può avere caratteristiche diverse da quelle di una macro se le è stato attribuito un significato mediante il comando primitivo `\let` o un comando equivalente. In quest'ultimo caso la sequenza di controllo, anche se ha tutto l'aspetto di una macro, può non essere affatto una macro: se per esempio mediante `\let` si rende `\miocomando` equivalente a `\newcommand`, `\miocomando` diventa un sinonimo di `\newcommand` e rappresenta quindi una macro; invece, se mediante `\let` si rende `\miaA` equivalente ad `A`, `\miaA` diventa un sinonimo di `A`, quindi rappresenta una lettera, non una macro; infatti in questo caso si dice che la sequenza di controllo è un *carattere implicito*.

Queste sono le situazioni delicate che talvolta rendono difficile la programmazione perché è facile confondersi; le spiegazioni diventano difficili perché la lingua italiana sembra poco flessibile per descrivere questo genere di cose. In relazione a quanto si diceva poco sopra, `\dots` è una macro; invece `\infty` ha le caratteristiche di una macro con pdfL^AT_EX, ma quelle

2.4. LA VIRGOLA INTELLIGENTE

di un carattere implicito con \LaTeX ; perciò la macro della virgola intelligente di questa classe deve saper distinguere le due situazioni sulla base del nome della sequenza di controllo, non solamente sul suo significato.

Non è assolutamente necessario inserire uno o più indici analitici in ogni guida tematica; dipende dall'argomento e dalla complessità di questi indici.

Di solito un solo indice analitico è sufficiente a meno che non si tratti di un indice di parecchie decine di pagine; in questo caso sarebbe forse meglio suddividere l'unico grosso indice analitico in indici parziali suddivisi per argomenti.

Questa classe viene distribuita con due file di stile per l'indice analitico: `guidatematica.ist` e `guidatematicafiles.ist`; entrambi gli stili dividono il loro contenuto in “sezioni” marcate con la lettera iniziale del gruppo di voci. Ma quando l'iniziale non è una lettera, le voci vengono raccolte sotto la definizione di “Simboli”; se l'iniziale fosse una cifra, sarebbe raccolti in una sezione intitolata “Numeri”. Ma se si facesse un indice analitico contenente solo nomi di file e delle loro estensioni (che cominciano con un punto) non sarebbe conveniente raccogliere le estensioni sotto una intestazione “Simboli”, ma è meglio raccoglierle sotto l'intestazione “Estensioni”; questo è quello che fa lo stile definito da `guidatematicafiles.ist`.

Per questo motivo in questa guida l'unico indice analitico è configurato con il seguente comando che usa la sintassi del pacchetto *imakeidx*:

```
\makeindex[options=%  
{-s guidatematica.ist},intoc]
```

Come si vede, l'argomento di `options` contiene la specificazione di usare lo stile `guidatematica.ist`, in modo che quando pdfLaTeX o XeLaTeX invia al sistema operativo il comando di eseguire `makeindex` gli passa anche questa opzione.

In questo indice analitico unico i file sono tutti elencati sotto la voce “file” quindi anche le estensioni stanno bene raggruppate sotto questa voce.

La definizione preimpostata nella classe è infatti:

```
\DeclareRobustCommand*\file[1]{%  
  \filestyle{#1}%  
  \index{file!#1@\filestyle{#1}}}
```

Ma se si volesse predisporre un indice dei file usati separato dall'indice analitico generico, allora bisognerebbe specificare:

```
\makeindex[name=files, options=%  
  {-s guidatematicafiles.ist},intoc,  
  title=Indice dei file]
```

e i nomi dei file o delle loro estensioni andrebbero specificati con il comando:

<code>\index[files]{\langle nome o estensione di un file \rangle}</code>
--

quindi il comando `\file` andrebbe ridefinito così:

```
\DeclareRobustCommand*\file[1]{%  
  \filestyle{#1}%  
  \index[file]{#1@\filestyle{#1}}}
```

Si noti che il comando `\DeclareRobustCommand` provvede a dichiarare il suo argomento come “robusto”; se il comando è già definito, si limita a mettere un avviso nel file `.log` e non produce un errore come fanno invece i comandi `\newcommand` o `\renewcommand`.

Alcuni utenti considerano molto utile avere l'indicazione delle voci inviate all'indice segnate nel margine della pagina dove la voce da indicizzare è riportata. Può essere utile per togliere i comandi “indicizzanti” dai luoghi dove una certa parola o una certa locuzione è usata, ma non si trova in una posizione importante all'interno del testo; in una posizione, cioè, nella quale si dica qualcosa di importante a proposito della parola o della locuzione. L'autore allora, con l'aiuto di queste indicazioni a margine può facilmente ritrovare le voci indicizzate e analizzare criticamente se quella voce vada riportata nell'indice con riferimento a quella specifica pagina, oppure se non sia importante e possa quindi essere omessa.

Per fare questo esisterebbe il pacchetto *showidx*, ma si raccomanda di non usarlo, perché la classe *memoir* ne provvede le funzionalità, anzi

CAPITOLO 3. GLI INDICI ANALITICI

le provvede meglio, nel senso che si possono attivare e disattivare diverse volte lungo il documento. Il comando per attivare questa funzionalità è `\showindexmarks` e quello per disattivarla è `\hideindexmarks`. Il compositore stia bene attento a disattivare tutte le istanze di attivazione quando compone la versione finale del documento.

Il pacchetto *imakeidx* nella versione 1.2e è reso perfettamente compatibile con la funzionalità della classe *memoir* e i due comandi di attivazione e disattivazione continuano a funzionare anche quando si usi questo utilissimo pacchetto.

pdfLaTeX, XeLaTeX o LuaLaTeX?

4

Chiaramente la scelta di uno dei tre programmi dipende esclusivamente dall'autore. Vale la pena di segnalare qui alcune differenze per consentire una scelta consapevole.

XeLaTeX e LuaLaTeX differiscono da pdfLaTeX, sotto alcuni aspetti sostanziali:

I FONT OPENTYPE

I font OpenType sono usabili solo con XeLaTeX e LuaLaTeX; veramente sarebbero usabili anche con pdfLaTeX ma sarebbe necessario creare ed installare opportuni file virtuali per spezzare le numerose pagine di glifi contenute nei file OpenType, in singole pagine di soli 256 caratteri e bisognerebbe ridefinire molti comandi di \LaTeX per accedere ai singoli frammenti di un font OpenType spezzato come detto sopra. Disponendo di XeLaTeX e LuaLaTeX¹, queste acrobazie con i font si possono evitare facilmente usando, appunto, XeLaTeX o LuaLaTeX.

IL FORMATO DI USCITA

Sia pdfLaTeX sia XeLaTeX sia LuaLaTeX producono l'uscita in formato PDF; il primo e il terzo producono quel formato in modo diretto, mentre il secondo passa attraverso una versione estesa e modificata del formato DVI e provvede ad eseguire la conversione in formato PDF mediante il programma esteso `xdvipdfmx`; siccome il formato DVI, seppure esteso, non consente alcune delle funzionalità del formato PDF, la conversione anche se ben fatta, non può introdurre nel file PDF quello che non c'era nel file DVI esteso. Passando, quindi, per

¹Con le distribuzioni del sistema \TeX successive al 2006 tutte contengono \XeLaTeX e quelle successive al 2009 contengono \LuaLaTeX ; questi motori di composizione sono disponibili per tutte le piattaforme di elaborazione Windows, Linux, Mac.

CAPITOLO 4. pdfLaTeX o XeLaTeX o LuaLaTeX?

XeLaTeX si perdono alcune funzionalità; LuaLaTeX, invece, consente di conservare tutte le funzionalità associate al formato PDF.

IL FORMATO PDF ARCHIVIABILE

Non so quanto sia importante per le guide tematiche del [GJIT](#), ma con XeLaTeX non si può produrre il file di uscita in formato PDF compatibile con le norme ISO 19005-1:2005 per l’archiviabilità; oggi poi esiste un aggiornamento della norma, ISO 19005-2:2011, e a maggior ragione il file PDF generato con XeLaTeX non è conforme con le norme ISO citate. Tuttavia ricorrendo a *pdfpages* è possibile ottenere la conversione in formato archiviabile in modo corretto; nella sezione di Documentazione del sito del [GJIT](#) esistono alcuni testi che descrivono il modo di procedere. Dal 2017 con LuaLaTeX è possibile produrre un file PDF conforme alle norme suddette; anzi, la conformità si ottiene più facilmente con LuaLaTeX che con pdfLaTeX.

LA MICROTIPOGRAFIA

pdfLaTeX e LuaLaTeX possono sfruttare ogni artificio di microtipografia per comporre le pagine in modo praticamente perfetto; con i font in dotazione, possono comporre sfruttando sia la protrusione di certi segni completamente o parzialmente nei margini destro e sinistro, sia l’espansione dei font, tecnica che permette di espandere o comprimere i caratteri di una riga di quel poco che basta per passare inosservato ma che rende gli spazi nella riga molto più omogenei fra righe adiacenti, cosicché non appaiono i brutti “ruscelli” bianchi che serpeggiano tra le parole attraverso la pagina mal composta. XeLaTeX può usare solo la protrusione dei segni, senza l’espansione dei caratteri. Molte delle raffinatezze che il pacchetto *microtype* permette di fare con i caratteri (per esempio spazieggiare il maiuscolo o il maiuscolo; definire e usare fattori di spazio diversi per i segni di interpunzione, ...) possono essere eseguite anche con oculate impostazioni delle proprietà dei font OpenType, ma il risultato finale della composizione è comunque migliore con l’uso dell’espansione dei caratteri.

I FONT CON I CORPI OTTICI

Il sistema T_EX è nato con i font corredati dei corpi ottici; fin dall’inizio esistevano i font Computer Modern a 5, 6, 7, 8, 9, 10, 12, 14,4, ... punti tipografici. Questo significa che al variare del corpo cambia anche un pochino il disegno dei segni in modo che i più piccoli siano ancora leggibili e i più grandi non appaiano troppo neri. I font

Computer Modern sono stati via via sostituiti da font in codifiche diverse ma contenenti più segni (da 128 si è passati a 256); oggi quelli più “moderni” sono i font Latin Modern accessibili normalmente da pdfLaTeX richiamando il pacchetto *lmodern* oppure, ancor meglio, il pacchetto *cfr-lm*; questo consente di selezionare sia font di forma e di serie diverse da quelle tradizionali, sia alcune particolarità non accessibili con il pacchetto *lmodern*. Quasi tutti gli altri font accessibili con pdfLaTeX sono dotati di un unico corpo e tutti gli altri corpi sono ottenuti per ingrandimento o rimpicciolimento proporzionale; il risultato è che spesso i corpi più piccoli sono quasi illeggibili perché i loro tratti sono diventati troppo sottili, e quelli più grandi di serie media ricordano più la serie nera che la serie media, visto che i tratti dei segni sono diventati troppo spessi rispetto allo spazio occupato da ogni singolo segno.

La situazione non è molto diversa con XeLaTeX e LuaLaTeX; la maggior parte dei font OpenType disponibili è dotato di un solo corpo. Fanno eccezione, ovviamente, i font OpenType Latin Modern, che consentono quindi di mantenere la massima leggibilità qualunque sia la dimensione dei caratteri. Se vogliamo, la gestione dei font Latin Modern è leggermente migliore sotto alcuni punti di vista rispetto a quella che si può avere con pdfLaTeX, ma sotto certi altri aspetti richiede un coinvolgimento maggiore da parte dell'autore. Esistono anche altri pochi font dotati di corpi ottici, e spesso il loro uso non è completamente libero come avviene per i Latin Modern.

Infatti per avere le proprietà dei font Latin Modern in formato Type 1 usabili da pdfLaTeX bastano poche righe di codice per caricarli e usarli:

- nel preambolo basta usare un solo statement con le caratteristiche desiderate; qui si è usato:

```
\RequirePackage[tt={%
    oldstyle=false,tabular,
    monowidth}]{cfr-lm}
```

Con questa unica riga si dispone dell'intera collezione con le impostazioni di default, tranne che per il font teletype per il quale non si vogliono le cifre minuscole (old style), ma si vogliono quelle maiuscole adatte alle tabelle, e si vuole la varietà di font

monospaziati.

- in questa classe sono anche già disponibili i comandi `\texttv` per selezionare il font teletype proporzionale e `\texttm` per usare quello a spaziatura fissa indipendentemente dalle impostazioni specificate fra le opzioni del pacchetto; inoltre il comando `\textl` garantisce l'uso delle cifre maiuscole con qualunque famiglia, forma e serie.

Per avere le stesse funzionalità con XeLaTeX e LuaLaTeX si sono dovute specificare le famiglie dei font: principale (font proporzionale con grazie e le cifre minuscole); sans serif (font proporzionale sgraziato o lineare, che dir si voglia, con le cifre minuscole); teletype proporzionale con le cifre maiuscole; teletype a spaziatura fissa con le cifre maiuscole; infine il maiuscoletto con le cifre minuscole.

Con queste impostazioni si è però ridefinito il comando `\textl` per le cifre maiuscole tenendo conto delle particolarità dei font OpenType, e si sono dovuti definire con `\DeclareTextFontCommand` i comandi equivalenti a quelli disponibili con il pacchetto *cfr-lm* usabile solo da pdfLaTeX.

```
\RequirePackage{fontspec}
\setmainfont[Ligatures=TeX,Numbers=OldStyle]{Latin Modern Roman}
\setsansfont[Ligatures=TeX,Numbers=OldStyle]{Latin Modern Sans}
\setmonofont[Numbers=Lining]{Latin Modern Mono}
\newfontfamily{\tvfamily}[Numbers=Lining]{Latin Modern Mono Prop}
\newfontfamily{\scfamily}[Ligatures=TeX,Numbers=OldStyle]{Latin
Modern Roman Caps}

\DeclareRobustCommand\scshape{\scfamily}
\DeclareTextFontCommand{\texttm}{\ttfamily}
\DeclareTextFontCommand{\texttv}{\tvfamily}
\DeclareTextFontCommand{\textsc}{\scfamily}
\def\textl#1{\addfontfeature{Numbers=Lining}#1}
```

In compenso è stato possibile specificare gli stessi font Latin Modern adatti alla matematica, in modo da sfruttare appieno le potenzialità dei font OpenType matematici; il pacchetto *unicode-math* mette a disposizione tutti i comandi matematici necessari.

```
\RequirePackage{unicode-math}
\setmathfont{Latin Modern Math}
```


VELOCITÀ DI COMPOSIZIONE

I tre programmi di composizione hanno velocità di composizione dello stesso testo molto diverse; il più veloce è pdfLaTeX; il più lento è LuaLaTeX; nonostante XeLaTeX produca un file DVI esteso e poi lo converta automaticamente in PDF, la sua velocità di composizione è intermedia rispetto agli altri due programmi. Il compositore di una guida tematica deve quindi tenere conto anche di questo fatto e scegliere il programma di composizione a ragion veduta. È ovvio che LuaLaTeX ha molte funzionalità in più rispetto agli altri due programmi, ma non vale la pena usarlo se non si ha veramente bisogno di quelle ulteriori funzionalità.

Con tali impostazioni comporre la documentazione con questa classe produce sostanzialmente gli stessi risultati che si usi pdfLaTeX, XeLaTeX oppure LuaLaTeX.

Perché scegliere l'uno o l'altro programma di composizione? Dipende dal contenuto della guida tematica; se occorre fare uso di alfabeti diversi da quello latino (per esempio: greco, cirillico, arabo, eccetera) allora è opportuno, se non obbligatorio, ricorrere a XeLaTeX o a LuaLaTeX, quest'ultimo solo nel caso che sia necessario servirsi delle sue maggiori funzionalità. Altrimenti è preferibile usare pdfLaTeX.

Questa guida composta con XeLaTeX e con LuaLaTeX presenta qualche piccola differenza nelle stesse pagine dovute al fatto che il motore LuaTeX sfrutta appieno le funzionalità del pacchetto *microtype*, compresa l'espansione dei font, che XeLaTeX non può usare, ma queste differenze non sono tali da giustificare da sole l'uso di LuaLaTeX; nello stesso tempo questa particolare guida non ha bisogno delle ulteriori funzionalità di LuaLaTeX, anzi, la sua composizione con pdfLaTeX è ottima e veloce con differenze trascurabili rispetto al prodotto di LuaLaTeX. Questo non vuol dire che con altre guide non sia preferibile agire diversamente.

INDICE ANALITICO

A

`\allineamento`, 19, 33
`\allineamentostyle`, 19, 31
`\amb`, 19, 21, 33
ambiente
 array, 19
 blockdescription, 20
 compactdescription, 20, 35
 compactenumerate, 20
 compactitemize, 20, 34
 description, 20, 35
 flushleft, 37
 minipage, 14, 31
 pdfxmetadata, 20
 plaindescription, 20, 35
 sintassi, 20, 37
 tabular, 19
 thebibliography, 37
 theindex, 37
 titlepage, 38
 VerbatimOut, 24
 Verbatim, 24
`\ambstyle`, 18, 19, 21, 31
`\Arg`, 17, 29
`\author`, 8, 12, 14, 38
`\authorspace`, 8, 12

B

`\backmatter`, 9, 15
`\Bambiente`, 16, 28
`\bibliography`, 9
`\bibsection`, 15

C

`\chapter`, 9
chiave
 before, 36
 font, 35
 format, 35, 36
`\chiave`, 19, 32
`\chiavestyle`, 18, 19, 31
`\cite`, 37
`\class`, 20, 33
classe
 beamer, 23
 book, 34
 guidatematica, 3, 4, 15, 27,
 29, 34
 memoir, 3, 10, 15, 22, 25, 34,
 38, 43, 44
`\classstyle`, 18, 20, 31
`\clearpage`, 9
`\comando`, 16, 28

INDICE ANALITICO

- \conta, 19, 32
- \contastyle, 18, 19, 31
- \cs, 16, 28

- D
- \Dambiente, 17, 28
- \date, 12, 14, 38
- \DeclareRobustCommand, 43
- \DeclareTextFontCommand, 48
- \def, 8, 12, 40
- \descrittore, 19, 32
- \descrittorestyle, 19, 31
- \documentclass, 38
- \documentclass, 22, 29
- \dots, 40

- E
- \Eambiente, 16, 28

- F
- file
 - .bbl, 16
 - .bst, 7
 - .dtx, 3
 - .eps, 24
 - .ist, 7, 8
 - .jpg, 24
 - .log, 21, 43
 - .mps, 24
 - .pdf, 24
 - .png, 24
 - .tex, 9
 - arstexnica.bst, 25
 - GuidaTematica-doc.pdf, 7
 - guidatematica-doc.tex, 29
 - guidatematica.bst, 16, 20, 25, 37
 - guidatematica.cls, 35
 - guidatematica.dtx, 7
 - guidatematica.ist, 20, 42
 - guidatematica.pdf, 7, 16
 - guidatematica.zip, 7
 - guidatematicafiles.ist, 20, 42
- \file, 20, 33, 36, 43
- \filedate, 14, 17, 29
- \filename, 17, 29
- \filestyle, 18, 20, 31, 35
- \fileversion, 14, 17, 29
- \frontmatter, 11, 15
- \frontmatter*, 11

- G
- \garg, 16, 28
- \GetFileInfo, 8, 12, 14, 15, 17, 29, 38

- H
- \hideindexmarks, 18, 21, 26, 44

- I
- \index, 32, 43
- \indicizzafile, 36
- \infty, 40
- \IntelligentComma, 39
- \item, 35, 36

INDICE ANALITICO

J

`\jobname`, 8

L

`\let`, 12, 40

`\licenza`, 12, 13, 17, 30

`\listoffigures`, 9

`\listoftables`, 9

M

`\mainmatter`, 15

`\mainmatter*`, 9, 11

`\makeindex`, 8, 18, 32

`\maketitle`, 9, 12, 14, 17, 30, 38

`\Marg`, 17, 29

`\marg`, 16, 28

`\meta`, 16, 28

`\misura`, 19

N

`\newcommand`, 12, 43

`\newpage`, 9

`\NoIntelligentComma`, 39

`\normalfont`, 28, 35

`\num`, 39

`\numeri`, 19, 32

`\numeristyle`, 18, 31

O

`\Oarg`, 29

`\oarg`, 16, 28

`\obeylines`, 37

`\opz`, 19, 32

opzione

9pt, 22

a4paper, 10, 22

a5paper, 10, 22

b5paper, 10, 22

ipertesto, 10, 11

italian, 26

oneside, 10, 22

pad, 10, 22

T1, 26

tablet, 10, 22

utf8, 26

`\opzstyle`, 18, 19, 31

P

pacchetto

afterpage, 27

babel, 26, 39

biblatex, 16, 25, 37

cfr-lm, 23, 26, 47, 48

color, 25

decorations, 25

enumitem, 3, 27, 34, 35, 37

etoolbox, 25

fancyvrb, 24, 27

fontenc, 26

fontspec, 26

glossaries, 11

graphicx, 24

guit, 24

hyperref, 11

hyperref, 11, 13

hypertext, 32

iftex, 23

imakeidx, 16, 18, 20, 25, 37,

42, 44

- inputenc*, 26
- listing*, 24
- listings*, 27
- lmodern*, 26, 47
- metalogo*, 24
- microtype*, 24, 46, 49
- multicol*, 25
- natbib*, 15, 16, 20, 25, 37
- pdfpages*, 46
- pict2e*, 24
- polyglossia*, 26, 27, 39
- showidx*, 25, 43
- siunitx*, 39
- tcolorbox*, 23
- TikZ*, 23
- tikz*, 25
- tkzexample*, 23
- unicode-math*, 48
- xcolor*, 25
- `\pack`, 20, 33
- `\packstyle`, 18, 20, 31
- `\posizione`, 19, 32
- `\posizionestyle`, 19, 31
- `\prebibhook`, 9, 15, 18
- `\printindex`, 9
- `\prog`, 20, 33
- programma
 - biber, 16
 - LuaLaTeX, 10, 23, 24, 26, 27, 45–49
 - LuaTeX, 49
 - makeindex, 42
 - pdfLaTeX, 10, 23, 24, 26, 42, 45–49
 - xdvipdfmx, 45
 - XeLaTeX, 10, 23, 24, 26, 27, 42, 45–49
 - `\progstyle`, 18, 20, 31
 - `\ProvidesClass`, 17, 29
 - `\ProvidesFile`, 8, 12, 14, 17, 29
 - `\ProvidesPackage`, 17, 29
- R
 - `\relax`, 17
 - `\renewcommand`, 15, 43
- S
 - `\Sambiente`, 17, 28
 - `\setcopymark`, 17, 29
 - `\setotherlanguages`, 27
 - `\showindexmarks`, 18, 21, 26, 44
 - `\stile`, 19, 32
 - `\stilestyle`, 18, 19, 31
 - `\subtitle`, 12, 14, 17, 38
- T
 - `\tableofcontents`, 9
 - `\texorpdfstring`, 32
 - `\textl`, 18, 48
 - `\texttm`, 18, 48
 - `\texttv`, 18, 48
 - `\textwidth`, 14
 - `\title`, 8, 12, 14, 38
 - `\tvfamily`, 18
- U
 - `\umisura`, 32
 - `\umisurastyle`, 18, 31
 - `\usetikzlibrary`, 25

INDICE ANALITICO

V

`\Verb`, [24](#)

`\VerbatimInput`, [24](#)

X

`\XeLaTeX`, [24](#)