

Università degli Studi di Milano

FACOLTÀ DI SCIENZE E TECNOLOGIE

DIPARTIMENTO DI INFORMATICA

GIOVANNI DEGLI ANTONI



CORSI DI LAUREA TRIENNALE IN

CORSO DI LAUREA

UN TEMPLATE REALIZZATO

CON **TYPST**

Relatore: Prof. Enrico Fermi  
Correlatore: Prof. Ezio Auditore da Firenze  
Correlatore: Prof. Francesco Bianchi

Elaborato Finale di:

Nome Cognome

Matr. Nr. 123456

ANNO ACCADEMICO 2025 – 2026

*Questo lavoro è dedicato ai miei genitori*

*“What I cannot create, I do not understand”*

*– Richard Feynman*

*“It’s not only powerful, but it’s also inadequate”*

*– Miller Puckette*

# Ringraziamenti

Questa sezione, facoltativa, contiene i ringraziamenti.

# Indice

<b>Ringraziamenti</b>	<b>iii</b>
<b>Indice</b>	<b>iv</b>
<b>1 Introduzione</b>	<b>1</b>
1.1 Il template .....	1
1.1.1 Impostazioni .....	1
1.2 I contenuti .....	2
1.3 Organizzazione della tesi .....	3
1.4 Stile e forma .....	4
<b>2 Stato dell'arte</b>	<b>6</b>
2.1 Risorse .....	6
2.2 Buone pratiche .....	7
2.3 Bibliografia e sitografia .....	8
<b>3 Tecnologie utilizzate</b>	<b>9</b>
3.1 Generalità .....	9
3.1.1 La scrittura WYSIWYG vs. WYSIWYM .....	9
3.1.2 Risorse e strumenti .....	10
3.2 Suggerimenti sull'uso di Typst .....	11
3.2.1 Riferimenti incrociati .....	11
3.2.2 Ritorni a capo .....	11
3.2.3 Spazi tra parole .....	12
3.2.4 Ambienti per scrivere codice .....	12
3.2.5 Figure .....	13
3.3 Bibliografia .....	13
3.3.1 Generalità .....	13
3.3.2 Strumenti .....	14
<b>4 Nome del Progetto</b>	<b>15</b>

4.1	Panoramica del progetto .....	15
4.2	Implementazione .....	15
<b>5</b>	<b>Test</b> .....	<b>16</b>
5.1	Protocollo .....	16
5.2	Risultati .....	16
5.3	Osservazioni .....	17
<b>6</b>	<b>Conclusioni</b> .....	<b>18</b>
6.1	Conclusioni .....	18
6.2	Sviluppi futuri .....	18
<b>A</b>	<b>Informazioni generali sull'attività di tirocinio</b> .....	<b>19</b>
A.1	Aspetti Burocratici (CdL Triennali) .....	20
A.2	Aspetti Pratici .....	20
A.3	Gestione del Codice .....	21
<b>B</b>	<b>Documenti da Produrre</b> .....	<b>22</b>
B.1	Riassunto .....	22
B.2	Presentazione .....	23
	<b>Bibliografia</b> .....	<b>25</b>

# Capitolo 1

## Introduzione

Questo documento ha una duplice funzione: da un lato mostra un esempio completo di tesi redatto in Typst e conforme allo standard PDF/A, e dall'altro contiene suggerimenti e risposte a domande frequenti poste dagli studenti. Se ne raccomanda, pertanto, un'attenta lettura.

### 1.1 Il template

Il template LaTeX è stato sviluppato, negli anni, dai membri del Laboratorio di Informatica Musicale (LIM) dell'Università degli Studi di Milano, in particolare da: Giorgio Presti, Luca Andrea Ludovico, Federico Avanzini, e Marco Tiraboschi. Questo template è un porting in Typst di quello LaTeX del LIM, la cui versione continuamente aggiornata è disponibile su Overleaf al seguente link:

<https://www.overleaf.com/read/hmffzxxzhhdqn>

È stato principalmente inteso per gli elaborati finali del corso di laurea triennale in Informatica Musicale, e poi esteso anche agli altri CdL del Dipartimento di Informatica, ma può essere riadattato anche per altri corsi cambiando i metadati nel preambolo. Nel resto del documento, dove non specificato, useremo il termine *tesi* nella sua accezione generica che include anche gli elaborati triennali.

#### 1.1.1 Impostazioni

Per quanto riguarda Typst, è possibile impostare i seguenti parametri:

```
#show: progetto.with(  
  university: "Università degli Studi di Milano",  
  unilogo: "../template/img/unimi.svg",  
  faculty: [Facoltà di Scienze e Tecnologie],
```

```
department: [  
  Dipartimento di Informatica \  
  Giovanni degli Antoni  
],  
cdl: [  
  Corsi di Laurea Triennale in \  
  Corso di Laurea  
],  
printedtitle: "",  
title: "Un template meraviglioso",  
typeofthesis: "Elaborato Finale",  
author: (  
  name: "Nome Cognome",  
  serial_number: "123456",  
) ,  
language: "it",  
supervisors: (  
  "Prof. Enrico Fermi",  
) ,  
cosupervisors: (  
  "Prof. Ezio Auditore da Firenze",  
  "Prof. Francesco Bianchi",  
) ,  
academicyear: ""  
)
```

(questi listati sono gli argomenti default). Tutti gli attributi sono sufficientemente chiari – tranne `printedtitle`:

- Esso il titolo che appare *al frontespizio*, nel PDF
- `title` invece è quello che appare *nei metadata* del file – se `printedtitle` è vuoto, allora è uguale a `title`

## 1.2 I contenuti

Alcune volte, le tesi possono avere forti connotazioni interdisciplinari. È però fondamentale ricordarsi che si tratta di lavori in area informatica, e questo aspetto deve emergere con chiarezza. Anche gli elaborati di contenuto più umanistico devono mostrare rigore scientifico e uno sforzo di formalizzazione nel loro svolgimento. In concreto, sono molto apprezzati schemi, tabelle, formalismi grafici e/o matematici, presenza di parti significative di codice (ove possibile).

Uno scritto di questa estensione parte sempre da un punto pregresso, tipicamente l'analisi dello stato dell'arte e della letteratura. È poi richiesto un

contributo personale (e originale, nel caso delle tesi magistrali) per giungere a una conclusione formulata dall'autore, eventualmente anche in contrasto con il pensiero corrente o con quanto ci si prefigge all'inizio dell'opera.

Le tre fasi devono emergere con chiarezza:

- analisi dello stato dell'arte e/o della letteratura;
- contributo personale alla ricerca;
- conclusioni raggiunte, ed eventuali sviluppi futuri.

Nel loro complesso queste tre componenti devono rendere evidente la *rilevanza* del lavoro, in termini di

- chiarezza nella definizione del problema, delle motivazioni e degli obiettivi della tesi;
- connessione con la letteratura (così da mostrare che la tematica oggetto della tesi è di interesse per la comunità scientifica);
- rilevanza della tematica nell'ambito delle discipline informatiche.

## 1.3 Organizzazione della tesi

La scelta di come strutturare un lavoro esteso, quale un elaborato finale o una tesi, non è semplice nè univoca, pertanto quelli sotto elencati vanno presi come suggerimenti generici e aggiornati alla propria situazione personale.



Figura 1: Il processo di trasformazione delle idee in testo [1].

Prima di cominciare a scrivere testo è fondamentale creare un indice dei contenuti organizzato in maniera gerarchica (tesi, capitoli, sezioni, sottosezioni, ecc.). Una volta ottenuto un indice soddisfacente, questo potrà essere riempito dal testo della tesi. Un approccio informatico alla questione è il seguente: scrivere una tesi significa partire da un grafo di idee [1] e arrivare a creare un albero di contenuti testuali (un *minimum spanning tree* del grafo delle idee!). L'albero rappresenta la struttura del testo, dove il nodo radice è la tesi stessa, i suoi figli sono i capitoli, ecc. Le foglie dell'albero rappresentano le sottosezioni di livello



più fine. Idealmente l'albero dei contenuti dovrebbe avere le seguenti caratteristiche:

- essere bilanciato;
- avere un'altezza  $h = 4$  o  $5$ ;
- essere un albero  $n$ -ario con  $2 \leq n \leq 7$

Anche i contenuti testuali di tutti i nodi allo stesso livello dell'albero dovrebbero essere il più possibile bilanciati (ad esempio, sarebbe sbagliato avere un capitolo di una pagina e un altro di 30 pagine). È tollerato uno sbilanciamento a favore del capitolo che descrive il proprio lavoro specifico e i contributi innovativi.

Limitatamente al secondo livello dell'albero (capitoli), un esempio puramente indicativo di scaletta per una tesi di natura sperimentale è quello che segue:

- Indice
- Introduzione
- Capitolo sullo stato dell'arte
- Capitolo sulle tecnologie utilizzate<sup>1</sup>
- Capitolo sul caso di studio o sul software realizzato
- Capitolo sui test effettuati
- Breve capitolo su conclusioni e sviluppi futuri
- Bibliografia ed eventuale Sitografia
- Eventuali appendici (ad esempio, listati completi di codice, manuale utente, dimostrazioni, ecc.)

Sarà dunque opportuno prevedere, nel capitolo introduttivo, un esplicito richiamo alla struttura del documento. Ad esempio: «Il presente lavoro è organizzato come segue: nel Capitolo 1 ...».

## 1.4 Stile e forma

Lo stile di una tesi scientifica deve essere esatto, chiaro, compatto, oggettivo [2].

*Esatto* vuol dire che ogni parola utilizzata significa solo ciò che deve esprimere. È opportuno cercare di evitare sinonimi per riferirsi a concetti importanti. Ove possibile si definiscano variabili (la frequenza  $f$ , il tempo  $n$ , ecc.) e le si

---

<sup>1</sup>Questo capitolo si può omettere se le sole tecnologie utilizzate sono strumenti standard come Python, JavaScript, C++, Matlab, ecc.

usino in maniera sistematica. Vanno evitate il più possibile espressioni vaghe e non quantitative («abbastanza grande», «praticamente tutti», «molto pochi», ...). Si usino i pronomi con parsimonia, meglio ripetere qualche sostantivo nel testo piuttosto che lasciare delle ambiguità.

*Chiaro* significa che la struttura e la forma devono essere funzionali a trasmettere l'informazione nel modo più immediato e comprensibile. I titoli di capitoli e sezioni devono essere illustrativi dei contenuti (è anche utile avere un breve testo all'inizio di un capitolo o una sezione, prima di passare alle sotto-sezioni). Il testo deve essere diviso logicamente in frasi, paragrafi, capoversi: sono preferibili paragrafi brevi, diretti e dichiarativi, limitando le subordinate e gli incisi, e i capoversi devono essere coerenti con la logica del discorso. L'uso di termini tecnici va limitato solo ai casi necessari. Anche gli acronimi devono essere usati con parsimonia, e vanno spiegati la prima volta che vengono usati.

*Compatto* significa che si deve scrivere solo ciò che è necessario scrivere. Bisogna evitare ripetizioni di concetti e osservazioni. Bisogna limitare descrizioni e dettagli che non siano necessari alla comprensione del discorso generale. In particolare vanno evitate osservazioni banali, che risultino ovvie al lettore. Tra una forma verbosa («in considerazione del fatto che...») e una equivalente più sintetica («perché...») si preferisca la seconda.

*Oggettivo* significa che ciò che si scrive deve essere privo di elementi soggettivi e di elementi che possono influenzare la valutazione. Vanno accuratamente evitate considerazioni personali («Questa idea mi è venuta per la prima volta in occasione...»), nonché affermazioni opinabili il cui giudizio sia lasciato al parere personale dell'autore. L'unico modo per supportare tali prese di posizione è fornire dati scientifici a riprova della propria tesi, o riferirsi in modo esplicito a un antecedente bibliografico («ipse dixit») citandone la provenienza in bibliografia. Infine, l'elaborato deve essere steso in forma impersonale. Frasi quali «Durante la mia esperienza ho approfondito i temi...» sono facilmente sostituibili con locuzioni quali «Durante la fase di analisi sono stati approfonditi i temi...».

Un uso appropriato di figure e tabelle è essenziale per supportare i punti appena discussi. Figure e tabelle devono avere delle didascalie autoesplicative (che non richiedano di leggere il testo principale per capirne il significato). Il testo principale deve però sempre contenere un riferimento alla figura o alla tabella («Come mostrato in Fig.1, ...»). L'uso di schemi grafici (uno schema a blocchi di algoritmo di elaborazione audio, uno schema concettuale di progetto software, ecc.) è particolarmente utile a supportare la chiarezza e la compattezza dell'esposizione.

# Capitolo 2

## Stato dell'arte

Una delle domande più ricorrenti da parte degli studenti è cosa si intenda per «stato dell'arte». Concretamente, si tratta di individuare la situazione corrente riguardo la tematica trattata nell'elaborato. A titolo di esempio, se il lavoro di tesi si concentra su un (presunto) innovativo algoritmo per suggerire i brani di una playlist, è necessario verificare che tale approccio sia veramente innovativo e, in ogni caso, studiare gli approcci alternativi già disponibili in letteratura.

Per ottenere tale risultato, è necessario condurre ricerche approfondite a livello bibliografico (testi di riferimento, articoli scientifici, ecc.) e implementativo (sul web, sugli store, ecc.) **prima** di iniziare il proprio lavoro.

### 2.1 Risorse

Riguardo gli articoli scientifici, si caldeggia l'uso del motore di ricerca specializzato Google Scholar.<sup>2</sup> Esistono, poi, numerosi repository che consentono di accedere gratuitamente a pubblicazioni scientifiche, tra cui ResearchGate,<sup>3</sup> Zenodo,<sup>4</sup> e SBA - Sistema Bibliotecario di Ateneo.<sup>5</sup> In particolare quest'ultimo permette agli studenti dell'Ateneo di accedere gratuitamente a molte pubblicazioni altrimenti accessibili solo a pagamento.

È importante riconoscere, nella grande mole di letteratura scientifica disponibile, i lavori più autorevoli e affidabili. Un primo elemento per orientarsi è che non tutte le sedi di pubblicazione sono uguali. In particolare una regola generale (per quanto non assoluta) è che gli articoli pubblicati su *rivista* sono tipicamente più completi e rigorosi di quelli pubblicati su atti di *convegno*. Un secondo elemento correlato all'autorevolezza della pubblicazione è il numero di citazioni

---

<sup>2</sup><https://scholar.google.it/>.

<sup>3</sup><https://www.researchgate.net/>

<sup>4</sup><https://zenodo.org/>

<sup>5</sup><http://www.sba.unimi.it/index.html> con contenuti scaricabili solo dalla rete interna all'ateneo.

che questa ha raccolto da parte di altri ricercatori (dato visibile ad esempio su Google Scholar). Infine, è opportuno riconoscere le sedi di pubblicazione (riviste e convegni) rilevanti per il settore: per questo, si consiglia di confrontarsi con i propri relatori e di consultare indicatori di qualità come Scimago JR<sup>6</sup> per le riviste e il GII-GRIN-SCIE Conference Rating<sup>7</sup> per le conferenze.

## 2.2 Buone pratiche

Alcuni consigli di buone pratiche per l'analisi dello stato dell'arte.

- Imparare a leggere in maniera *efficace*. Ci sono migliaia di articoli potenzialmente interessanti, quindi è essenziale riuscire a estrarre in breve tempo gli elementi rilevanti di un articolo senza perdersi nei dettagli (fino a che non sia strettamente necessario), nonché annotare in maniera ordinata tali elementi.
- Imparare a leggere in maniera *critica*. Raramente un articolo scientifico è perfetto, gli aspetti tecnici e gli eventuali punti deboli (metodologie, ripetibilità dei risultati, ecc.) vanno valutati e annotati con attenzione.
- Imparare a seguire le «piste» interessanti. Una volta trovato un articolo rilevante, è utile esaminare
  - gli articoli che esso cita (i suoi riferimenti bibliografici): questo permette di rintracciare riferimenti autorevoli a cui l'articolo si appoggia;
  - gli articoli che lo citano (ad esempio Google Scholar offre questa funzionalità): questo permette di rintracciare riferimenti più recenti che hanno proseguito nella stessa direzione di ricerca.
- Imparare a lavorare iterativamente.
  - Aggiungere alla propria bibliografia gli articoli considerati rilevanti, a mano a mano che vengono trovati.
  - Raggrupparli iterativamente in sottotematiche.
  - Usare inizialmente un approccio «inclusivo» (nel dubbio, aggiungere un articolo in bibliografia piuttosto che scartarlo), e solo in un secondo tempo decidere cosa tenere e cosa scartare.

---

<sup>6</sup><https://www.scimagojr.com>

<sup>7</sup><https://scie.lcc.uma.es:8443>

## 2.3 Bibliografia e sitografia

La bibliografia di un lavoro scientifico deve necessariamente essere ricca. Tutti i testi in bibliografia devono essere citati almeno una volta nel corso dell'elaborato.

Nella valutazione della bibliografia da parte della commissione, i testi cartacei sono considerati significativamente più validi dei siti consultati. Ne consegue che la bibliografia debba essere ricca di testi pubblicati, siano essi libri, articoli, al limite tesi di laurea o di dottorato o rapporti tecnici. Si suggerisce di evitare citazioni a fonti di dubbia valenza scientifica, quali Wikipedia, W3Schools, ecc.

Se è necessario citare siti Web, esistono tre strade ugualmente accettabili:

1. se il numero di siti non è preponderante rispetto ai testi «tradizionali», è possibile inserirli parimenti in bibliografia;
2. se i siti da citare sono numerosi, è più opportuno creare una sorta di bibliografia parallela e separata, detta *sitografia*;
3. infine, se la citazione dei siti serve a individuare un prodotto e non una fonte di informazioni, la soluzione più opportuna è quella delle note a piè di pagina.

Al momento, 12-06-2025, Typst **non** supporta le bibliografie multiple – esiste il package alexandria [3] per sopperire a questa mancanza. In futuro il template potrà fornire delle funzioni ad-hoc per integrare questa funzione.

# Capitolo 3

## Tecnologie utilizzate

In questo capitolo vengono presentati alcuni suggerimenti utili per un utente Typst alle prime armi.

### 3.1 Generalità

#### 3.1.1 La scrittura WYSIWYG vs. WYSIWYM

L'acronimo WYSIWYG sta per «What You See is What You Get», e si riferisce al concetto di ottenere sulla carta testo e immagini che abbiano una disposizione grafica equivalente a quella visualizzata a schermo dal software di videoscrittura. Un esempio classico di WYSIWYG è Microsoft Word, che mostra il testo impaginato e formattato come ci si aspetta di vederlo una volta stampato.

L'acronimo WYSIWYM sta per «What You See is What You Mean», ed è il paradigma per la creazione di testi strutturati. Typst è un ambiente che supporta tale paradigma. In realtà, anche Microsoft Word avrebbe la possibilità di strutturare il testo, principalmente attraverso il meccanismo degli stili, ma pochissimi utenti sfruttano tale funzionalità (ovviamente se sceglierete di scrivere la tesi in Word raccomandiamo caldamente l'uso di tali funzioni, così come le funzioni di gestione automatica dei riferimenti e della bibliografia).

I principali svantaggi di un sistema WYSIWYM sono il tempo di apprendimento, dovuto a una minore intuitività degli strumenti software, e la necessità di invocare la compilazione del documento per vederne l'aspetto definitivo. Ad esempio, in Typst l'intero documento viene scritto in testo semplice, che all'interno contiene ambienti e comandi con informazioni di layout, e solo la compilazione permette di scoprire eventuali errori di sintassi e giungere, infine, alla creazione del PDF.

Le difficoltà iniziali, però, sono ampiamente compensate dai vantaggi a medio e lungo termine. Infatti, il lavoro risulterà perfettamente impaginato e struttura-

to, e dunque avrà un aspetto professionale. Questo riguarda non solo gli stili, che vengono applicati al testo in modo coerente con il template prescelto, ma anche aspetti tipicamente spinosi di Word, quali il posizionamento delle immagini e delle tabelle, la creazione di una bibliografia con relative citazioni nel testo, la creazione di un sommario. Diventa automatico e molto semplice, ad esempio, aggiungere un indice delle figure o delle tabelle, oppure numerare le formule espresse nel testo. Un altro aspetto su cui Typst è nettamente superiore a Word è proprio la scrittura di formule matematiche, come mostrato nell'esempio qui riportato:

$$x_i(n) = a_{i1}u_1(n) + a_{i2}u_2(n) + \dots + a_{iJ}u_J(n).$$

### 3.1.2 Risorse e strumenti

Esiste una vastissima gamma di risorse online per avvicinarsi a Typst. Un buon punto di partenza è navigare l'applicazione web di Typst,<sup>8</sup> che contiene una ricca documentazione su come funziona il linguaggio,<sup>9</sup> un archivio di librerie e template<sup>10</sup> e anche un forum nel quale trovare le domande più frequenti.<sup>11</sup>

In alternativa a un'installazione locale sul proprio pc, è possibile utilizzare un editor Typst online, con il vantaggio di avere immediatamente a disposizione l'ambiente di sviluppo e tutti i package necessari, nonché di potere condividere il proprio progetto con il relatore di tesi. Il più diffuso editor online per questo linguaggio è l'app web fornita direttamente da Typst: una semplice iscrizione tramite mail permette di accedere all'editor.

Qualunque sia la risorsa utilizzata, ecco un elenco non esaustivo di argomenti di base nei quali con tutta probabilità ci si imbatte durante la stesura della tesi.

- Formattazione del testo (grassetto, italics, dimensioni font, ecc.) e del documento (heading di vario livello, indici, ecc.).
- Elenchi: ambienti *enum* e *list*.
- Riferimenti incrociati: comando *ref*, *label* ed etichette.
- Matematica: equazioni e modalità *inline*.
- Figure: formati grafici, ambiente *figure* e *image*.
- Tabelle: ambienti *table* e *grid*.

---

<sup>8</sup><https://typst.app/>

<sup>9</sup><https://typst.app/docs/>

<sup>10</sup><https://typst.app/universe/>

<sup>11</sup><https://forum.typst.app/>

- Riferimenti e bibliografie (si veda più sotto la sezione Sezione 3.3).

## 3.2 Suggerimenti sull'uso di Typst

Fatte salve le indicazioni generali fornite nella sezione precedente, di seguito si riportano alcune osservazioni puntuali sulle domande e gli errori più tipici degli studenti alle prime armi con Typst.

### 3.2.1 Riferimenti incrociati

Uno dei principali vantaggi di Typst è la possibilità di impostare riferimenti automatici a molti elementi del documento, tra cui headings, tabelle, figure, equazioni, riferimenti bibliografici, e via dicendo.

Quindi il modo corretto per riferirsi, ad esempio, alla seconda sezione non è scrivere «Sezione 2» bensì «Sezione 2». Il risultato apparente (nel PDF) è lo stesso, mentre ci sono differenze sostanziali a livello di codice. Il vantaggio è che, se il secondo capitolo diventasse il terzo, il riferimento incrociato continuerebbe a puntare alla posizione corretta. Si pensi, per estensione, alla numerazione delle immagini, o ai riferimenti alla bibliografia.

Sintatticamente, questo richiede di inserire delle label `<mia_label>` dopo una *figure* che contiene gli elementi cui ci si vuole riferire, e dei comandi `#ref(<mia_label>)` o `@mia_label` dove si vuole creare il riferimento. La sintassi con `@` si utilizza anche per riferirsi agli elementi della bibliografia (si veda più sotto la sezione Sezione 3.3 per la generazione di una bibliografia completa).

### 3.2.2 Ritorni a capo

I ritorni a capo in Typst possono essere effettuati in due modi: con la sintassi `\` o con una doppia pressione del tasto di ritorno a capo. In generale, la soluzione corretta è la seconda, che equivale a usare il tasto Enter in Word. Il Backslash, che corrisponde a Shift+Enter in Word, crea una nuova riga senza interruzione del paragrafo. Questo va usato solo in casi molto specifici, come nella frase seguente.

Il sito web ufficiale dell'Università degli Studi di Milano è:

`https://www.unimi.it`

In questo template, un nuovo paragrafo (dopo un doppio a capo) crea un rientro della prima riga. Questo viene fatto impostando il parametro `first-line-indent` di par al valore `1.2em`. Non c'è nulla di male nel rientro, ma se proprio lo si vuole evitare la soluzione è rimuovere il parametro appena citato.



### 3.2.3 Spazi tra parole

Riguardo la gestione della spaziatura tra parole, Typst adotta una strategia molto elegante, che lascia uno spazio maggiorato dopo il punto di fine periodo. Un potenziale problema è che questo spazio extra viene introdotto dopo qualsiasi occorrenza del punto, indipendentemente dalla funzione sintattica, e dunque anche dopo i nomi puntati, quali «R. Schumann», o dopo le formule «ad es.», «Fig. n», «ecc.» e via dicendo. Per evitarlo, questi spazi da non aumentare vanno sostituiti con alternative, quali un Backslash seguito da uno spazio (che immette un *control space*) o una tilde ~ (che introduce un *unbreakable space*, utile a impedire ritorni a capo intermedi).

### 3.2.4 Ambienti per scrivere codice

Il codice all'interno dell'elaborato va scritto con carattere monospaziato e rispettando, nell'ambito del possibile, le originali regole (o buone pratiche) di indentazione.

Per farlo, esistono all'interno del Typst universe una serie di librerie per la scrittura di codice, come `lovelace` o `algo`.

Una soluzione che non attinge ai repository Typst è aprire e chiudere un blocco con il triplice accento ``` e inserire al suo interno il codice da renderizzare.

Un esempio, relativo al calcolo del massimo comun divisore attraverso l'algoritmo di Euclide in Python, è:

```
def MCD(a,b):  
    while b != 0:  
        a, b = b, a % b  
    return a
```

Se dopo l'apertura del blocco si specifica il linguaggio adottato, ad esempio con la sintassi ````python`, si ottiene automaticamente l'evidenziazione del codice:

```
def MCD(a,b):  
    while b != 0:  
        a, b = b, a % b  
    return a
```

L'elenco dei linguaggi ufficialmente supportati non è disponibile, ma è presente una lista<sup>12</sup> non ufficiale creata da un utente grazie all'autocompletamento dell'editor della web app.

---

<sup>12</sup><https://github.com/typst/typst/issues/1511#issuecomment-1822459773>

### 3.2.5 Figure

In tutti i casi in cui sia possibile (schemi a blocchi, plot di dati, ecc.), è opportuno che le figure siano in formato vettoriale (svg) per aumentarne la leggibilità. Nel caso di figure prodotte da software esterno (ad esempio, grafici esportati in eps o pdf da Matlab), è consigliabile conservare tutti i sorgenti e i dati utilizzati per generarle: in questo modo sarà possibile ricreare le figure quando necessario.

Le figure devono sempre avere riferimenti nel testo, realizzati come visto nella Sezione 3.2.1. Il medesimo discorso vale anche per Tabelle ed Equazioni.

Si evitino espressioni del tipo «come visibile nella figura seguente» in favore di riferimenti esatti del tipo «come visibile in Fig Figura 1» in quanto Typst posiziona le immagini sulla pagina seguendo regole tipografiche, che non corrispondono necessariamente alla posizione di inserimento nel sorgente del documento.

## 3.3 Bibliografia

### 3.3.1 Generalità

Esistono più modi per inserire una bibliografia in Typst: o quello di BibLaTeX [4] oppure Hayagriva [5]. Entrambi consentono di aggiungere, rimuovere e modificare voci di bibliografia in maniera efficiente, di formattarle, di riordinarle a piacere e aggiornare automaticamente i corrispondenti riferimenti nel testo, ecc.

La differenza è che il primo è lo standard che arriva da LaTeX – e quindi tutto ciò che ne consegue: ampio supporto, letto ed esportato ovunque – mentre il secondo è l’approccio alla Typst, quindi molto più agile, intuitivo; ma in quanto più moderno non si aspetti lo stesso supporto.

**Per BibLaTeX** Una guida introduttiva e completa è «Tame the BeaST».<sup>13</sup> In estrema sintesi, i passi per gestire una bibliografia tramite sono essenzialmente tre.

1. Salvare i riferimenti bibliografici come entry di uno o più file con l’estensione `.bib` (si veda ad esempio il file `bibliografia.bib`, parte di questo template). Gli entry sono scritti in un formato specifico, in particolare ogni entry ha una propria etichetta testuale che lo identifica univocamente

---

<sup>13</sup>Accessibile da <http://www.tug.org/interest.html>

2. Creare la bibliografia alla fine del documento o dove desiderato, usando il comando `#bibliography("file.bib")`. È possibile inoltre specificare uno stile bibliografico modificando il campo `style` del comando precedente.
3. All'interno del testo, riferirsi a una voce di bibliografia tramite il comando `@etichetta_entry`. Si noti che una voce bibliografica non viene inclusa in bibliografia in assenza di una citazione all'interno del testo (coerentemente con quanto discusso nella sezione Sezione 2.3).

Tutte queste considerazioni si applicano anche ad Hayagriva.

La scelta del sistema ricade sulla fonte della bibliografia: se è principalmente da scrivere *a mano*, allora conviene sicuramente Hayagriva perché è molto più semplice; altrimenti, se la bibliografia viene recuperata (o generata) da fonti online è meglio BibLaTeX perché è un formato standard e supportato da qualsiasi software.

È consigliabile infine cominciare a costruire la propria bibliografia a mano a mano che si analizza lo stato dell'arte, invece che rimandare alla stesura finale della tesi.

### 3.3.2 Strumenti

Un file `.bib` è un file di testo e può quindi essere gestito con un qualsiasi text editor. Esistono comunque molti tool più evoluti per gestire bibliografie in formato bib. Un'applicazione installabile localmente sul proprio pc è JabRef.<sup>14</sup> Oppure esistono tool online, come Zotero,<sup>15</sup> che forniscono molte funzionalità tra cui l'esportazione di bibliografie in formato bib.

Peraltro, anche Google Scholar esporta automaticamente citazioni in formato bib cliccando sul link Cita (icona con doppie virgolette) e scegliendo l'opzione nella parte bassa della finestra che si apre. **Attenzione** però: spesso i bib esportati da Scholar sono incompleti o sporchi, è sempre consigliabile controllarne la correttezza.

Infatti, si preferiscono, in generale, i metadati raccolti dalla sorgente primaria della risorsa (es. sito ufficiale della pubblicazione) rispetto a quelli presentati dai motori di ricerca. Nel caso in cui i siti ufficiali non propongano il formato, esistono convertitori, facilmente reperibili e utilizzabili via web, per convertire in la maggior parte dei formati (es. RIS). Nei rari casi in cui non siano disponibili i dati in alcun formato machine readable, potete compilare manualmente l'entry voi stessi.

---

<sup>14</sup><http://www.jabref.org>

<sup>15</sup><http://www.zotero.org>

# Capitolo 4

## Nome del Progetto

In genere il capitolo più corposo dell'elaborato è quello in cui si parla del lavoro svolto. Esistono alcune buone pratiche per rendere l'esposizione efficace, eccone alcune.

### 4.1 Panoramica del progetto

Prima di addentrarsi nei dettagli è bene fornire una panoramica (anche molto schematica, corredata da un diagramma) del lavoro svolto, in modo che il lettore abbia una mappa concettuale con cui orientarsi.

### 4.2 Implementazione

Una volta fornita una panoramica, è possibile addentrarsi nei dettagli, ricordando che si sta scrivendo un articolo scientifico e non un testo di narrativa. Vanno dunque evitate domande retoriche quali *«ma come è stato possibile risolvere questo problema? Ebbene ...»*.

Vale la pena riportare nel testo solo le parti di codice più importanti, demandando all'appendice il codice completo e altri extra come il manuale utente.

# Capitolo 5

## Test

Ogni lavoro scientifico richiede una validazione dei risultati ottenuti. Questo si può fare confrontando in modo sistematico il proprio lavoro con lavori concorrenti o misurando l'efficacia del lavoro mediante test con gli utenti. È fondamentale che questi test siano ripetibili, dovrete dunque fornire tutti i dettagli necessari nel testo per permettere a chi legge la tesi di replicare l'esperimento.

Progettare e condurre un test soggettivo con utenti è un lavoro complesso e lungo, che richiede pianificazione e competenza. Il bel libro di Lazar *et al.* [6] illustra in dettaglio i concetti principali della ricerca sperimentale e le metodologie correlate: ipotesi di ricerca, design sperimentale, analisi dei risultati sperimentali. Quelli che seguono sono alcuni consigli specifici sugli aspetti più importanti.

### 5.1 Protocollo

Uno degli errori più comuni è sottovalutare lo studio di un *protocollo* sperimentale. Affinché i risultati dei test siano significativi è necessario non trascurare i seguenti aspetti:

- eliminare distorsioni sistematiche involontarie;
- isolare le variabili oggetto dello studio;
- garantire una numerosità sufficiente del campione;
- confrontare gli effetti con un gruppo di controllo.

### 5.2 Risultati

I risultati dei test vanno presentati in modo chiaro e completo, possibilmente indicando la significatività statistica di quanto ottenuto.

È buona norma fornire sia i dati numerici (un esempio di come si fanno le tabelle in Typst è visibile in Tab. Tabella 1), sia una rappresentazione grafica (a barre, a scatole e baffi, a violino, di dispersione, ecc.).

È inoltre consigliato riportare in appendice i dati grezzi completi, in modo da permettere al lettore di ripetere eventuali test statistici.

Colonna 1	Colonna 2	Colonna 3
5	8	1
6	9	2
7	0	3

Tabella 1: Tabella di esempio.

### 5.3 Osservazioni

Quando si traggono conclusioni dai dati bisogna prestare attenzione a non confondere la correlazione con un rapporto di causalità. Molto spesso accade che un test suggerisca la presenza di un fenomeno, ma non dica nulla sulla causa. In questo caso bisogna formulare delle ipotesi, calcolare le implicazioni, ed eseguire un test che valuti se e quali di queste implicazioni si verifichino. Se il nuovo test falsifica la teoria, non importa quanto questa sia elegante: è falsa. Se invece il nuovo test non falsifica la teoria, allora la si può dare per «vera fino a prova contraria».

Per queste ragioni è necessario esporre le proprie osservazioni in maniera cauta, senza andare oltre ciò che suggeriscono i dati. È certamente possibile speculare sulle cause, ma va esplicitato chiaramente, e tali speculazioni vanno supportate dalla bibliografia.

# Capitolo 6

## Conclusioni

### 6.1 Conclusioni

Nelle conclusioni si tirano le somme di quanto realizzato, facendo un riassunto stringato del lavoro svolto. In particolare vanno dichiarati punti di forza e criticità della ricerca effettuata, nonché quali aspetti dello stato dell'arte siano stati superati dal lavoro in oggetto.

### 6.2 Sviluppi futuri

Tra gli sviluppi futuri in genere trovano posto quelle migliorie non realizzate per mancanza di tempo, la cui necessità è emersa solo dopo i test, e che riguardano il progetto ad un livello di astrazione più alto (nel caso di tesi che si inquadrano in una linea di ricerca).

# Appendice A

## Informazioni generali sull'attività di tirocinio

Lo studente che si appresta a svolgere il lavoro di tesi generalmente incontra relatore e correlatore a inizio progetto per definire una tabella di marcia, quindi al completamento di ogni obiettivo fissato si ripresenta per un aggiornamento e un controllo. Questo è molto importante per approcciare il problema in modo organico e per evitare di illudersi di aver raggiunto un risultato sufficiente ritenendo concluso il lavoro.

Durante il lavoro è consigliato di prendere appunti su ciò che si sta facendo, in modo da avere una base per la stesura dell'elaborato.

Infine, è richiesto agli studenti di consegnare la proposta di indice prima di iniziare la stesura, per poi consegnare ogni capitolo a mano a mano che questi vengono completati. La tesi completa va consegnata per la rilettura finale a relatore e correlatore almeno una settimana prima della consegna ufficiale, per permettere un ultimo *proof reading* e l'integrazione delle eventuali modifiche richieste.

Altri dettagli:

- E' estremamente importante che lo studente comunichi con i relatori per aggiornamenti regolari, in particolare ogni volta che ci sono sviluppi significativi o decisioni da prendere.
- E' altresì fondamentale che ogni comunicazione via mail avvenga mettendo sempre tutti i relatori/correlatori tra i destinatari.
- Di norma non serve venire a lavorare in laboratorio, tuttavia se lo studente lo preferisce è il benvenuto. Per ottenere il badge di accesso si acceda al portale: <https://badge.di.unimi.it/>



## A.1 Aspetti Burocratici (CdL Triennali)

Solitamente l'attività di tirocinio interno per la triennale si articola nei seguenti passaggi formali:

1. Scegliere un argomento su cui lavorare: è possibile partire da una propria idea o scegliere un tema tra quelli proposti dai docenti
2. Scegliere relatore e correlatore: il relatore deve essere uno strutturato del Dipartimento (Ricercatore, Professore associato o Professore ordinario)
3. Definire insieme ai relatori: titolo (anche provvisorio), obbiettivi e tabella di marcia
4. Procedere con l'apertura formale del tirocinio sulla piattaforma <https://tirocini.di.unimi.it>
5. Il tirocinio deve durare almeno 14 settimane. Durante questo tempo si dovrebbe svolgere il grosso del lavoro
6. A tirocinio ultimato, se e solo se il relatore è d'accordo, si procede con la chiusura formale del tirocinio. Questa consiste da un lato nell'iscrizione all'appello apposito sul SIFA/UniMIA a carico dello studente, e dall'altro nell'inoltro della richiesta di chiusura a carico del relatore
7. Iscrizione alla sessione di laurea sul sito del Corso di Laurea  
[https://<nome\\_cdl>.cdl.unimi.it/it/studiare/laurearsi](https://<nome_cdl>.cdl.unimi.it/it/studiare/laurearsi)  
(es. <https://informaticamusicale.cdl.unimi.it/it/studiare/laurearsi>)
8. Consegna di elaborato e riassunto in formato PDF-A (questo template produce già il formato richiesto) non superiore a 10MB
9. Discussione pubblica, che in quanto tale è aperta a spettatori esterni
10. Festeggiamenti (si ricorda che questi devono essere svolti nel rispetto dei luoghi e delle istituzioni rappresentate)

## A.2 Aspetti Pratici

Relativamente agli aspetti pratici, ogni relatore ha un suo *modus operandi* per la gestione dei tesisti, quanto riportato di seguito va quindi considerato a solo titolo esemplificativo; in particolare questa è la procedura tipica seguita dagli studenti del Corso di Laurea di Informatica Musicale:

1. Analisi dello stato dell'arte (Cap. Sezione 2).  
E' importante annotare tutto sin da subito usando Typst, perché poi andrà scritto nell'elaborato.
2. Definizione del problema, «design» del sistema, scelta degli strumenti
3. Implementazione
4. Definizione ed esecuzione della campagna di test e/o validazione (Cap. Sezione 5)
5. Definizione dell'indice dell'elaborato
6. Scrittura dell'elaborato e correzione (meglio se in parallelo con gli altri passi)
7. Consegna dell'elaborato
8. Stesura della presentazione, correzione, suggerimenti

### **A.3 Gestione del Codice**

Nella maggior parte dei tirocini dei CdL in Informatica, lo studente si troverà a scrivere del codice. Si consiglia di concordare in fase preliminare il modo di condivisione del codice. Per quanto riguarda il LIM, il codice pertinente alle tesi viene generalmente versionato su GitHub (<https://github.com/LIMUNIMI>).

Nel caso di utilizzo di git (con qualsiasi remoto, sia esso GitHub, GitLab, Bitbucket, ...), si consiglia allo studente di avere una infarinatura generale prima di utilizzarlo per il proprio progetto. Un buon tutorial interattivo è <https://learngitbranching.js.org>, anche se include argomenti avanzati che possono essere al di là delle necessità contingenti dello studente.

# Appendice B

## Documenti da Produrre

I membri del LIM in generale non richiedono agli studenti di consegnare una copia cartacea del proprio lavoro. In altri casi, si invita lo studente a concordare questo punto con il proprio relatore. In prossimità della discussione viene richiesto loro di condividere in formato digitale:

- la versione finale dell'elaborato;
- il riassunto caricato in piattaforma;
- il codice sviluppato;
- tutti i materiali di supporto utilizzati e/o prodotti;
- la presentazione predisposta per la discussione.

La consegna può avvenire via chiavetta USB o attraverso i sistemi di condivisione dei file (WeTransfer, Dropbox, Google Drive, ecc.).

### B.1 Riassunto

In prossimità della data di laurea, viene richiesto agli studenti di caricare in piattaforma anche il riassunto. La sua funzione è permettere alla commissione di visionare rapidamente gli argomenti trattati da ciascun elaborato.

L'estensione di tale documento non dovrebbe superare le due pagine, e spesso una pagina è più che adeguata allo scopo. Riguardo i contenuti, il riassunto riprende – in maniera molto succinta – lo schema dell'elaborato finale, e quindi, ad esempio, la scaletta proposta in Sezione 1.3.

Visto lo scopo del documento, è buona prassi bilanciare il testo in modo da dedicare poche righe allo stato dell'arte, in favore delle parti originali dell'elaborato.

## B.2 Presentazione

Gli studenti delle lauree triennali hanno a disposizione al massimo 10 minuti per presentare il proprio elaborato, che diventano 15 per le lauree magistrali.

Si raccomanda di provare ripetutamente il discorso cronometrando e considerando i tempi morti per l'accensione del PC, l'avvio dei software, il passaggio da un'applicazione a un'altra, e via dicendo. Si segnala inoltre la possibilità di prenotare l'aula della discussione per provare in anticipo i collegamenti e verificare che immagini e audio vengano correttamente riprodotti.

Per via degli stringenti vincoli temporali, difficilmente una presentazione multimediale può superare la soglia di 15 slide. Questo valore è puramente indicativo, perché dipende dalla densità di informazioni della slide e dal tempo dedicato a ciascuna slide mentre si sta presentando.

La scelta tecnologica più comune è Microsoft PowerPoint, ma qualsiasi alternativa (Prezi, Acrobat PDF, Typst, ecc.) è ammissibile.

Gli errori più comuni da evitare sono:

- slide dal contenuto testuale troppo ricco;
- slide che verranno lette parola per parola in fase di presentazione.

Entrambi i problemi si risolvono impostando il testo in maniera estremamente schematica, tendenzialmente per elenchi puntati. Si presti attenzione, però, a usare una forma omogenea. Ad esempio, sarebbe poco elegante il seguente elenco, che include aggettivi, sostantivi e frasi di senso compiuto, e non fa un uso corretto di iniziali maiuscole:

- applicativo esteticamente molto gradevole;
- C'è la possibilità di interagire con altri utenti;
- l'utente può esportare file in formato XML.

Una possibile revisione, basata solo sui sostantivi, è:

- qualità estetica;
- interazione tra utenti;
- supporto dell'XML.

Un altro consiglio per rendere la presentazione cognitivamente più gestibile da parte del pubblico è quello di aggiungere in calce ad ogni slide un indicatore dello stato di avanzamento (ad esempio «Slide 3 di 15»). In questo modo, in caso di esaurimento del tempo concesso la commissione saprà se dare modo di

concludere («Slide 14 di 15») o se fermare il candidato («Slide 14 di 138»). In assenza di un indicatore la commissione tende a presupporre il secondo caso.

Per quanto riguarda gli argomenti da trattare durante la discussione, è opportuno esporre brevemente lo stato dell'arte per concentrarsi sul proprio lavoro, sui test e sui risultati. Una tecnica per capire quanto tempo dedicare a ogni argomento consiste nel dare un peso a ciascuna parte di cui si vuole parlare, quindi assegnare un numero di minuti proporzionale a detto peso in modo da raggiungere il tempo massimo a disposizione.

Video dimostrativi e *live demo* sono gradite, ma è sconsigliabile dedicare troppo tempo a questi contributi, a meno che al contenuto multimediale si sovrapponga una spiegazione da parte del candidato.

# Bibliografia

- [1] W. Härmäläinen, «Scientific Writing for Computer Science Students».
- [2] W. Strunk, *Elements of Style*. Penguin, 2007.
- [3] «Package Alexandria». [Online]. Disponibile su: <https://typst.app/universe/package/alexandria>
- [4] «BibLaTeX cheat sheet». [Online]. Disponibile su: <https://tug.ctan.org/info/biblatex-cheatsheet/biblatex-cheatsheet.pdf>
- [5] «Bibliografia con Hayagriva». [Online]. Disponibile su: <https://github.com/typst/hayagriva>
- [6] J. Lazar, J. H. Feng, e H. Hochheiser, *Research methods in Human-Computer Interaction*, Second. Cambridge: Morgan Kaufmann, 2017.
- [7] G. Haus e M. Longari, «A multi-layered, time-based music description approach based on XML», *Computer Music Journal*, vol. 29, fasc. 1, pp. 70–85, 2005.
- [8] F. Mittelbach, M. Goossens, J. Braams, D. Carlisle, e C. Rowley, *The LaTeX Companion*, Second. Boston: Addison-Wesley, 2004.
- [9] «LIM official website».



Progetto sviluppato presso il laboratorio ADAPT LA  
<https://cazzola.di.unimi.it/adapt-lab.html>