

ALMA MATER STUDIORUM · UNIVERSITÀ DI BOLOGNA

SCUOLA DI SCIENZE
Corso di Laurea Triennale in Informatica

**REDESIGN E SVILUPPO
DI UN'APPLICAZIONE WEB
PER LA FILOLOGIA MODERNA**

**Relatore:
Chiar.mo Prof.
FABIO VITALI**

**Presentata da:
GUGLIELMO GUALERZI**

**Sessione I
Anno accademico 2019/2020**

Questa pagina è lasciata intenzionalmente bianca.

Keywords:

Informatica umanistica
Edizioni critiche digitali
TEI e Markup
Extreme Programming
Participatory design

Indice

1	Introduzione	1
2	Filologia moderna	7
2.1	Filologia	7
2.2	Edizioni critiche	9
2.3	Edizioni digitali	10
2.4	Edizioni critiche digitali	11
3	Strumenti digitali al servizio della filologia	13
3.1	Edizioni precedenti di PhiloEditor	13
3.1.1	Funzionalità di PhiloEditor	13
3.1.2	Limiti di PhiloEditor (edizioni passate)	20
4	PhiloEditor 7	23
4.1	Tipi di Utenza	23
4.2	Funzionalità del sistema	24
4.2.1	Home	25
4.2.2	Introduzione	28
4.2.3	Opere	29
4.2.4	Modalità di visualizzazione	31
4.2.5	Laboratorio	34
4.2.6	Ricerca sulle opere	36
5	Nuova struttura: PhiloEditor 7	39
5.1	Tecnologie	39

5.1.1	TNTSearch	40
5.1.2	wikEd diff	40
5.2	Front-end	41
5.2.1	Caricamento iniziale	41
5.2.2	Navigazione	41
5.2.3	Diffing	41
5.2.4	Modalità di visualizzazione	42
5.2.5	Ricerca nel testo	43
5.2.6	Statistiche	43
5.2.7	Caricamento dei documenti	43
5.2.8	Salvataggio delle marcature	43
5.2.9	Esportazione TEI	44
5.2.10	Ricerca nelle opere	44
5.2.11	Sicurezza	45
5.3	Back-end	46
5.3.1	Richieste sui documenti	46
5.3.2	Ricerca nelle opere	47
5.3.3	Statistiche	49
5.3.4	Salvataggio delle marcature	49
5.3.5	Condivisione delle marcature	50
5.3.6	Registrazione	50
5.3.7	Login / logout	51
5.3.8	Aggiornamento della password	51
6	Metodologia di sviluppo software	53
6.1	Extreme Programming	53
6.2	Participatory design	55
7	Valutazioni	59
7.1	Efficienza	59
7.2	Efficacia	60
8	Conclusioni e sviluppi futuri	63

Questa pagina è lasciata intenzionalmente bianca.

Capitolo 1

Introduzione

L'obiettivo di questa tesi è quello di migliorare uno strumento digitale semplice ed intuitivo dedicato all'analisi filologica delle opere letterarie. Questo strumento ha preso forma nell'applicazione web chiamata PhiloEditor.

PhiloEditor cerca di creare un connubio tra informatica e filologia, fornendo un'ambiente ricco di tool dedicati al filologo che desideri avvalersi di mezzi digitali per velocizzare e migliorare i suoi studi.

L'applicazione si fonda sul concetto di poter mostrare al ricercatore le differenze che intercorrono nel tempo tra le edizioni della stessa opera. Questo punto di partenza si pone infatti come imprescindibile, secondo la definizione stessa di filologia moderna:

«La filologia moderna si articola in due grandi direzioni: da una parte si tende a reperire, ricostruire e interpretare i testi (studio delle testimonianze verbali); e dall'altra a mettere in luce e interpretare fatti di ogni genere che giovino alla comprensione dei testi stessi (studio delle cose). Nel primo suo aspetto la f. cerca di reperire documenti letterari ancora ignoti; di accertare l'autenticità sia di questi sia di quelli già noti; di ricostruire il documento nel suo aspetto originario, depurandolo dalle alte-

razioni prodottesi in esso nel corso dei secoli per effetto della trasmissione a opera di intermediari: copisti, stampatori, editori (filologia testuale). Inoltre, la filologia cerca di interpretare i testi ricostruiti, facendo tesoro delle necessarie cognizioni linguistiche, metriche, stilistiche e di quelle storiche, allo scopo di rivivere la scrittura nelle medesime condizioni culturali del suo autore; d'individuare le fonti del documento letterario e la fortuna di esso. Nell'altro aspetto la f. studia la biografia dell'autore, le condizioni storiche e sociali del suo tempo e del suo ambiente particolare.» [1]

PhiloEditor sfrutta perciò un sistema che permette di visualizzare in maniera chiara e stratificata le differenze fra un'edizione più vecchia ed un'edizione più recente della stessa opera. Tutto questo per fornire al filologo una visione di come il testo si sia evoluto nel tempo, passando da una versione iniziale ad una più evoluta e corretta. Tale analisi fornisce anche al ricercatore la possibilità di comprendere meglio il testo e ricavare informazioni sulle decisioni e le motivazioni che hanno spinto l'autore ad effettuare le modifiche alla sua opera.

Tuttavia PhiloEditor non è solamente uno strumento di visualizzazione, infatti esso si pone anche l'obiettivo di dare al moderno filologo la possibilità di editare le proprie edizioni critiche digitali.

«... (PhiloEditor è) un ambiente digitale per la rappresentazione e la marcatura delle modifiche temporali di testi letterari per l'allestimento di edizioni critiche digitali. L'obiettivo della piattaforma è evoluto dalla semplice caratterizzazione filologica dell'evoluzione temporale di testi letterari ad un approccio critico ed ermeneutico di percorsi interpretativi.» [2]

«In PhiloEditor le marcature tipografiche e cromatiche, solitamente impiegate per individuare differenti testimoni, sono applicate a diverse categorie variantistiche. Queste ultime possono essere distinte in categorie correttive e metodologie correttive.

La combinazione della visualizzazione stratigrafica con la tassonomia variantistica permette quindi di ottenere una descrizione caratterizzante in senso sincronico e diacronico.» [3]

Attualmente PhiloEditor fornisce la possibilità di visualizzare, studiare e marcare due opere: *I promessi sposi*, nelle edizioni “ventisettesima” (1827) e quarantana (1840), e *Le avventure di Pinocchio*, nelle edizioni del 1883 (Rivista) e del 1983 (Volume).

PhiloEditor ha visto il succedersi di varie edizioni, giungendo fino alla versione 6.5. Tale edizione fornisce già le basi sopracitate ed anche altri tool.

In particolare esso fornisce un sistema di visualizzazione e di marcatura delle opere, oltre alla possibilità di effettuare il login, salvare i propri documenti marcati, di visualizzarne le statistiche ed esportarli in formato TEI.

Tuttavia l'applicazione presenta alcuni problemi.

Prima fra tutte l'interfaccia, di difficile comprensione per un neofita. L'interfaccia mostra infatti svariate lacune, quali la mancanza di un'appropriata suddivisione in sezioni del sito, la quale non permette una graduale ed intuitiva immersione nell'ambiente.

La disposizione non ottimale dei tool e la mancanza di una vera sezione dedicata all'introduzione all'utilizzo dell'applicazione rendono difficile la fruizione da parte di un pubblico che non possieda particolari competenze nella navigazione web.

Da notare inoltre che la piattaforma non è predisposta all'utilizzo da parte di un pubblico che non sia già competente nell'ambito della filologia, tagliando fuori così una discreta quantità di potenziali utenti. Inoltre vanno aggiunti alcuni bug nell'ambito delle statistiche, del caricamento della pagina, il quale tende molte volte a bloccarsi o ad imporre tempi d'attesa molto lunghi, e nella visualizzazione di una singola edizione, nella quale vengono mostrate comunque le varianti.

In merito ai tempi d'attesa da notare anche la suddivisione in tomi e non in capitoli de I promessi sposi, la quale si rivela poco efficiente, sia da un punto di vista della fruizione del contenuto che da quello del caricamento del testo, dalle tempistiche non indifferenti.

Infine PhiloEditor 6.5 manca di un'appropriata funzione di ricerca fra le opere, che fornisca al filologo informazioni riguardo alle occorrenze di parole chiave all'interno delle varie edizioni dei capitoli. Così imponendo una ricerca basata sulla continua apertura e chiusura di ciascun capitolo.

Tramite uno stretto dialogo con la committenza e l'applicazione del metodo dell'Extreme Programming (adattato in base alle dimensioni ridotte del team), è stato possibile venire incontro alle esigenze dei filologi, risolvendo molte delle problematiche sopraelencate ed aggiungendo nuovi tool e modalità di visualizzazione.

Questo lavoro ha preso forma in PhiloEditor 7.

L'applicazione è stata totalmente rinnovata.

Si è partiti dalla totale ricostruzione della veste grafica per renderla più intuitiva e mobile friendly, concordata con la committenza e modificata a seconda delle nuove esigenze che emergevano durante l'implementazione, il tutto reso possibile da un continuo e diretto dialogo.

Inoltre sono stati risolti i bug che affliggevano l'edizione passata, rendendo la piattaforma più veloce ed efficiente, ed è stata creata la modalità lettura che al contempo risolve i bug riguardanti la visualizzazione di una singola opera e fornisce una nuova modalità per approcciarsi all'opera.

È stata inoltre implementata una barra di ricerca nelle opere, capace di fornire e visualizzare le occorrenze delle parole chiave in tutte le edizioni di tutti i capitoli di tutte le opere presenti, così da permettere uno studio filologico ad un livello più alto rispetto al semplice confronto tra capitoli.

PhiloEditor 7, rispetto alle edizioni passate, dunque si presenta più in-

tuitivo e fruibile, adatto anche a chi non abbia particolari competenze tecniche, sia nell'informatica che nella filologia, aprendo dunque le porte anche ai neofiti e potenziando gli strumenti votati ai più esperti.

PhiloEditor 7 si pone certamente come una buona base di partenza, ma certamente potrà in futuro essere espanso, aggiungendo nuove funzionalità.

Ad esempio si potrebbe dotare l'utente della capacità di creare le proprie personali categorie correttive o creare una sezione per le esercitazioni, dato che adesso il sistema è fruibile da parte di utenti che non siano studenti universitari o ricercatori. Si potrebbero anche potenziare alcune funzionalità già esistenti, come la possibilità di un'utenza multilivello, e automatizzare alcuni meccanismi, quali il caricamento delle opere da parte di un utente di livello amministratore.

È plausibile che queste ulteriori espansioni e potenziamenti potrebbero permettere a PhiloEditor di divenire un'applicazione con una larga scala di utenza, essendo capace di fornire un valido supporto allo studio filologico avanzato e al contempo di introdurre neofiti allo studio della materia.

Questa pagina è lasciata intenzionalmente bianca.

Capitolo 2

Filologia moderna

PhiloEditor è un'applicazione che ha lo scopo di fornire uno strumento digitale per la filologia. In questo capitolo dunque verrà approfondito il concetto di filologia e di edizione critica, così da poter comprendere al meglio le basi su cui PhiloEditor si fonda.

2.1 Filologia

Secondo il filosofo, poeta, scrittore e filologo Friedrich Nietzsche:

«Filologia è quella onorevole arte che esige dal suo cultore soprattutto una cosa, trarsi da parte, lasciarsi tempo, divenire silenzioso, divenire lento, essendo un'arte e una perizia da orafi della parola, che deve compiere un finissimo attento lavoro e non raggiunge nulla se non lo raggiunge lento. Ma proprio per questo fatto è oggi più necessaria che mai; è proprio per questo che essa ci attira e ci incanta quanto mai fortemente, nel cuore di un'epoca del "lavoro": intendo dire della fretta, della precipitazione indecorosa e sudaticcia, che vuol "sbrigare" immediatamente ogni cosa (...). Per una tale arte non è tanto facile sbrigare qualsiasi cosa perché essa ci insegna a leggere bene, cioè a leggere lentamente in profondità, guardandosi avanti

e indietro, non senza secondi fini, lasciando porte aperte, con dita e con occhi delicati.» [4]

La definizione esatta di filologia è stata spesso oggetto di dibattito ed è mutata più di una volta nel tempo.

Secondo l'accezione comune attuale, la filologia è un'unione di discipline letterarie atte allo studio di testi di diverse tipologie¹, sia antichi che moderni. Tutto questo per ricostruire la loro forma originaria, tramite un'analisi critica e comparativa delle varie fonti disponibili riguardanti le opere oggetto di studio, così infine, tramite varie metodologie di indagine, da poter giungere ad un'interpretazione il più possibile corretta.

La figura dedita allo studio di questa disciplina è quella del filologo.

«Il termine filologo, derivato dal latino *philōlogus*, presso gli antichi Greci e Latini, indicava lo studioso amante della dottrina e dell'erudizione, in particolare storico-letteraria.» [5]

Attualmente il filologo può essere definito come la figura che studia i testi nella loro evoluzione storica, ricercandone modifiche e cambiamenti. Negli studi del filologo si rivela necessario perciò identificare il significato originale del testo, le intenzioni dell'autore e il contesto storico, culturale e politico in cui l'opera è stata scritta.

Il filologo deve quindi conoscere in modo approfondito storia e lingua della civiltà nel contesto di studio, e deve avere accesso a fonti e documenti delle epoche in esame, per poter restaurare in modo corretto l'interpretazione di un passo, una parola o un'opera, argomentando la scelta fatta.

Gli studi filologici si dividono in varie branche, solitamente dedicate al periodo storico e alla cultura da cui provengono i testi. Le due branche più studiate sono la filologia classica e la filologia moderna.

«La filologia classica è la scienza che studia i documenti scritti in lingua greca o latina, in vista della retta edizione e della

¹Letterari, storici, politici, economici, giuridici, ecc.

corretta interpretazione di essi nel contesto dell'epoca e della civiltà che li ha prodotti.» [6]

Nell'ambito della filologia classica, l'obiettivo del filologo è quello di individuare e ricostruire il testo nel formato originale, inizialmente individuandone un archetipo. Compito che si rivela spesso arduo, in quanto i mezzi di trasmissione della parola scritta delle epoche indagate spesso non sono stati in grado di sopravvivere al tempo. Di conseguenza, difficilmente è possibile avere la certezza che il testo preso in esame sia l'edizione originale e più antica.

Il filologo deve dunque cercare di evidenziare le linee correttorie dell'autore attraverso la catalogazione delle varianti².

Per quanto riguarda la filologia moderna, invece, essa si prepone lo scopo dell'analisi di testi più recenti, sia storicamente che culturalmente.

Questo determina un vantaggio da parte del filologo moderno rispetto al filologo classico, in quanto il filologo moderno avrà a disposizione versioni ed edizioni certe del testo, e non dovrà perciò passare dalla costruzione di un archetipo iniziale che fornisca l'ossatura dell'opera oggetto d'indagine.

L'obiettivo del filologo moderno è dunque quello di immedesimarsi nell'autore, nell'epoca e nel contesto politico, storico e sociale che lo hanno formato, così da poter comprendere al meglio i motivi che hanno spinto tale autore ad effettuare modifiche al suo testo, dette varianti .

2.2 Edizioni critiche

«...edizione critica: costituzione di un testo, in genere antico, fatta mediante un accurato confronto tra i manoscritti ed eventuali stampe esistenti, con l'intento di eliminare gli errori dovuti alla trasmissione e ricondurlo, per quanto possibile, alla sua forma genuina...» [7]

²Le varianti vanno dalla modifica di una o poche parole, fino ad intere porzioni dell'opera stessa.

Più in particolare, nell'ambito filologico,

«L'edizione critica di un testo letterario è il risultato di una serie di operazioni condotte con metodo scientifico, ossia verificabile e dimostrato, che mirano a stabilire, secondo l'ipotesi più economica, la forma del testo in oggetto più vicina possibile alla volontà dell'autore. Il concetto di testo è un concetto dinamico, dal momento che esso subisce variazioni e modifiche, sia da parte dell'autore sia da parte dei copisti durante la trasmissione nel corso del tempo. Le procedure di ricostruzione critica avvengono in due momenti: il primo documentativo, con la raccolta di tutte le testimonianze esistenti del testo stesso, dirette (manoscritti, stampe) e indirette (citazioni presenti in altre opere); il secondo interpretativo, nel quale, attraverso procedimenti differenziati a seconda se si tratta di testi antichi o di testi moderni e contemporanei, si fissa il testo e si procede, ove necessario, al restauro testuale.» [8]

Si può dunque notare come come l'edizione critica, se ben realizzata, possa fornire un buon livello di garanzia, ma non una verità assoluta, in quanto è il risultato di un'indagine sempre soggetta a possibili variazioni o modifiche. Come qualunque atto scientifico resta confutabile e alterabile di conseguenza.

Una buona edizione critica è capace di rivolgersi ad un ampio pubblico, dal filologo al lettore comune. Infatti da essa dipende la qualità e l'originalità del testo che si andrà a leggere, spesso molto diverso tra le varie edizioni. L'edizione critica ha dunque lo scopo di fornire una lettura quantomai più vicina al messaggio ed ai valori che l'opera cerca di trasmettere nella sua forma originale.

2.3 Edizioni digitali

«Un'edizione digitale scientifica (EDS) è una risorsa informatica che offre (solitamente) un'interpretazione critica di documenti

storici e di testi. Le edizioni digitali scientifiche non sono soltanto pubblicazioni in formato digitale; sono piuttosto sistemi di informazione che seguono una metodologia determinata da un paradigma digitale – così come le edizioni tradizionali a stampa seguono una metodologia che corrisponde ai paradigmi della cultura cartacea.» [9]

«...gli strumenti digitali permettono un allargamento del campo, una visione d'insieme, un'osservazione sintetica dell'ampia rete che lega i diversi testi, ma anche l'avvicinamento estremo all'oggetto di studio, l'analisi ravvicinata se non microscopica, attraverso la riproduzione ad altissima risoluzione dell'originale o addirittura l'analisi multi-spettrale, che permette l'accesso a ciò che il supporto nasconde nelle pieghe di una storia complessa e travagliata... » [3]

2.4 Edizioni critiche digitali

Un'edizione critica digitale unisce le caratteristiche e i punti chiave di un'edizione critica e quelli di una digitale.

Su questo concetto si basa PhiloEditor, il quale cerca di fornire la possibilità al filologo di realizzare e condividere le proprie edizioni critiche con strumenti digitali.

Questa pagina è lasciata intenzionalmente bianca.

Capitolo 3

Strumenti digitali al servizio della filologia

3.1 Edizioni precedenti di PhiloEditor

PhiloEditor nasce come strumento filologico, per aiutare e velocizzare l'operato degli studiosi in questo ambito.

La piattaforma ha preso forma in varie edizioni arrivando fino alla versione 6.5.

In questa evoluzione PhiloEditor ha accumulato svariate funzionalità che lo rendono una valida alternativa per un filologo interessato ad utilizzare uno strumento digitale dedicato alla sua ricerca.

3.1.1 Funzionalità di PhiloEditor

Il nucleo di PhiloEditor consiste nella possibilità di utilizzare un sistema di diff. Il diff è uno strumento capace di evidenziare le differenze tra due file, ovvero un software in grado di confrontare due testi ed evidenziarne le differenze, per poi organizzarle e mostrarle all'utente.

Piuttosto che essere character-oriented, il diff è line-oriented, e si pre-mura di restituire un output in un formato standard tale che sia umani che

elaboratori siano in grado di comprendere le differenze e possano “applicarle”: dato uno dei due file e le variazioni è possibile ricostruire il secondo file applicando le variazioni sul primo.

Solitamente il diff viene utilizzato per evidenziare le differenze fra due versioni dello stesso file, in particolare per file di codice, dove ogni linea di codice viene paragonata e ne vengono mostrate le differenze.

Nel caso di PhiloEditor, poiché l’analisi dev’essere effettuata su testi letterari e non righe di codice, si utilizza il diff sulle singole parole. In particolare ci si è avvalsi della libreria `wikiEd diff`, inizialmente creata per analizzare i Wikipedia source text, ma che ben si adatta a qualsiasi tipo di testo, sia a livello di parole che di caratteri [10]. Tramite una successiva analisi delle differenze trovate da `wikiEd diff` è stato inoltre possibile accorpere inserimenti e cancellazioni ravvicinati. Tutto questo per poter mostrare all’utente in maniera chiara ed immediata le differenze individuate.

Diversamente da una tradizionale interfaccia di diffing (Figura 3.1), in cui le due versioni vengono mostrate una accanto all’altra e accanto ad esse le differenze individuate, PhiloEditor utilizza un’approccio alternativo, basato sulla sovrapposizione (Figura 3.2): solo il testo dell’edizione meno recente viene mostrato e, dove si presentano delle variazioni, il testo viene evidenziato e ad esso vengono sovrapposte le porzioni modificate.

```

original:                                     new:
1 This part of the                            1 This is an important
2 document has stayed the                     2 notice! It should
3 same from version to                       3 therefore be located at
4 version. It shouldn't                      4 the beginning of this
5 be shown if it doesn't                    5 document!
6 change. Otherwise, that                   6
7 would not be helping to                   7 This part of the
8 compress the size of the                   8 document has stayed the
9 changes.                                   9 same from version to
10                                            10 version. It shouldn't
11 This paragraph contains                   11 be shown if it doesn't
12 text that is outdated.                   12 change. Otherwise, that
13 It will be deleted in the                 13 would not be helping to
14 near future.                              14 compress the size of the
15                                            15 changes.
16 It is important to spell                 16
17 check this dokument. On                   17 It is important to spell
18 the other hand, a                         18 check this document. On
19 misspelled word isn't                    19 the other hand, a
20 the end of the world.                    20 misspelled word isn't
21 Nothing in the rest of                    21 the end of the world.
22 this paragraph needs to                   22 Nothing in the rest of
23 be changed. Things can                   23 this paragraph needs to
24 be added after it.                       24 be changed. Things can
                                              25 be added after it.
                                              26
27 This paragraph contains                   27 This paragraph contains
28 important new additions                    28 important new additions
29 to this document.                         29 to this document.

The command diff original new produces the following normal diff output:

0a1,6
> This is an important
> notice! It should
> therefore be located at
> the beginning of this
> document!
>
11,15d16
< This paragraph contains
< text that is outdated.
< It will be deleted in the
< near future.
<
17c18
< check this dokument. On
---
> check this document. On
24a26,29
>
> This paragraph contains
> important new additions
> to this document.

Note: Here, the diff output is shown with colors to make it easier to read. The
diff utility does not produce colored output; its output is plain text. However, many
tools can show the output with colors by using syntax highlighting.

```

Figura 3.1: Visualizzazione tradizionale di un sistema di diffing.

CAPITOLO 1

Come andò che Maestro Ciliegia, falegname, trovò un pezzo di legno, che piangeva e rideva come un bambino

C'era una volta...

"Un re!" diranno subito i miei piccoli lettori.

"No, ragazzi, avete sbagliato". C'era una volta un pezzo di legno.

Non era un legno di lusso, ma un semplice pezzo da catasta, di quelli che d'inverno si mettono nelle stufe e nei caminetti per accendere il fuoco e per riscaldare le stanze.

Non so come andasse, ma il fatto gli è che un bel giorno questo pezzo di legno capitò nella bottega di un vecchio falegname, il quale aveva nome mastro Antonio, se non che tutti lo chiamavano maestro Ciliegia, per via della punta del suo naso, che era sempre lustra e paonazza, come una ciliegia matura.

Appena maestro Ciliegia ebbe visto quel pezzo di legno, si rallegrò tutto e dandosi una fregatina di mani per la contentezza, borbottò a mezza voce:

"Questo legno è capitato a tempo: voglio servirmene per fare una gamba di tavolino".

Detto fatto, prese subito l'ascia arrotata per cominciare a levargli la scorza e a digrossarlo, ma quando fu lì per lasciare andare la prima ascia, rimase col braccio sospeso in aria, perchè sentì una vocina sottile sottile, che disse raccomandandosi:

"Non mi picchiar tanto forte!"

Figuratevi come rimase quel buon vecchio di maestro Ciliegia!

Girò gli occhi smarriti intorno alla stanza per vedere di dove mai poteva essere uscita quella vocina e non vide nessuno! Guardò sotto il banco, e nessuno; guardò dentro un armadio che stava sempre chiuso, e nessuno; guardò nel corbello dei trucioli e della segatura, e nessuno; aprì l'uscio di bottega per dare un'occhiata anche sulla strada, e nessuno. O dunque?...

Figura 3.2: Visualizzazione sovrapposta applicata da PhiloEditor.

PhiloEditor fornisce anche la possibilità all'utente di individuare le categorie correttive nei testi, ovvero la propria personale interpretazione delle varianti, basate sullo studio dell'autore e del contesto storico e sociale che hanno visto la nascita delle varie edizioni. Essendo queste categorie correttive un'analisi interpretativa e non oggettiva, il sistema dà la possibilità a

ciascun utente di marcare nel testo le proprie categorie.

Previa autenticazione o iscrizione, l'utente può analizzare il testo e marcare le parole che ritenga appartenere alla categoria correttoria scelta. Per maggiore semplicità di lettura, le marcature vengono visualizzate tramite la colorazione dello sfondo delle parole scelte con lo specifico colore della categoria. PhiloEditor infatti fornisce per ogni opera una serie di categorie correttorie precaricate, ed a ciascuna è dedicato uno specifico colore (Figura 3.3), questo comporta che tutte le marcature fatte appartenenti alla stessa categoria avranno lo stesso colore di sfondo.

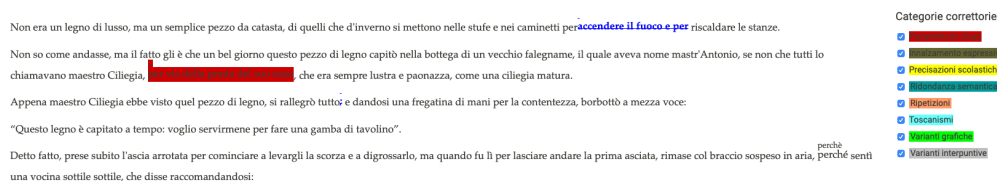


Figura 3.3: Esempio di marcatura del testo.

In seguito a questo processo di marcatura, l'utente può decidere se salvare il testo marcato e poi, se lo desidererà, potrà condividere con altri la sua personale marcatura.

Una volta effettuate le marcature e salvato il documento, è possibile visualizzare le statistiche. Le statistiche si presentano sotto forma sia numerica che con diagrammi a torta. Tutto questo per facilitare il filologo nella sua analisi, permettendogli di avere una chiara visione sul numero di caratteri coinvolti e sulla quantità di interventi, oltre che a permettergli di confrontare facilmente la sua marcatura con quelle altrui. Inizialmente PhiloEditor includeva come opera esclusivamente I promessi sposi di Alessandro Manzoni, nelle edizioni del 1827 e del 1840. Attualmente è stata aggiunta anche Le avventure di Pinocchio di Carlo Collodi, edizioni del 1883 e del 1983. Mentre per I promessi sposi si è optato per una suddivisione per tomi, Le avventure di Pinocchio è stato separato per capitoli. Da un menù ad albero della navbar è possibile selezionare il tomo o il capitolo da visualizzare.

Entrambe le opere vengono mostrate nella stessa interfaccia ed è possibile passare tra quattro opzioni di visualizzazione:

1. Edizione meno recente: viene mostrato esclusivamente il testo dell'edizione meno recente;
2. Edizione più recente: viene mostrato esclusivamente il testo dell'edizione più recente;
3. Verticale: viene mostrato il testo della versione meno recente e ad esso vengono sovrapposte le varianti dell'edizione più recente;
4. Orizzontale: viene mostrato il testo della versione meno recente e ad esso vengono poste accanto, segnalate mediante parentesi quadre, le varianti dell'edizione più recente.

Accanto al testo viene inoltre mostrato l'elenco delle varianti e delle categorie correttive. Queste quattro modalità permettono allo studioso di visualizzare i testi delle opere a seconda delle necessità. Ed è inoltre possibile, se si ha effettuato il login, tramite il bottone “modifica”, passare alla modalità di editing in cui marcare le opere (Figura 3.4).

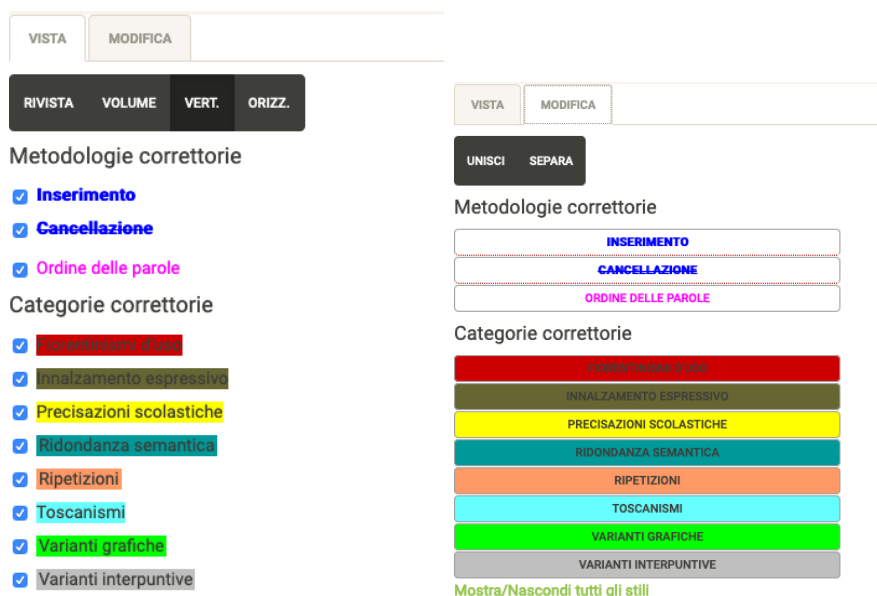


Figura 3.4: Confronto fra gli strumenti in modalità Vista e modalità Modifica

Con il susseguirsi delle versioni di PhiloEditor è stata aggiunta anche la possibilità di esportare i file tramite il sistema stabilito da TEI.

«The Text Encoding Initiative (TEI) is a consortium which collectively develops and maintains a standard for the representation of texts in digital form.» [11]

«Le linee guida TEI forniscono raccomandazioni sui modi di rappresentazione adeguati, in grado di mettere in evidenza determinate caratteristiche di un testo, al fine di facilitarne l'elaborazione su un computer, indipendentemente dalla piattaforma utilizzata, descrivendo uno schema di codifica che può essere espresso in diversi linguaggi formali. Sono esplicitamente orientate alla creazione e allo scambio di informazioni testuali, ma considerano anche altre tipologie di informazioni (quali immagini e suoni). In particolare, vengono specificati una serie di marcatori (tag) che possono essere inseriti nel testo, al

fine di contrassegnarne la struttura con le caratteristiche di interesse. Lo schema di codifica o linguaggio di markup indica l'insieme completo di norme connesse con l'uso di un linguaggio di marcatura in un determinato contesto. Lo schema di codifica TEI è particolarmente utile per facilitare lo scambio di dati tra utenti o gruppi di ricerca che utilizzano programmi e sistemi informatici diversi. Contiene un inventario con le caratteristiche più utilizzate per rappresentare testi. Le linee guida si applicano a qualsiasi tipo di testo, indipendentemente dalla data e dal genere letterario, senza restrizioni sulla forma o contenuto.» [12]

Nel caso di PhiloEditor il file marcato viene esportato con estensione XML. Nel nuovo file così creato vengono individuate tramite tag tutte le diverse varianti e categorie selezionate.

Infine, all'interno dell'applicazione web, è stata fatta l'aggiunta di una barra di ricerca nella navbar capace di effettuare ricerche all'interno del capitolo (o del tomo) che sta venendo in quel momento visualizzato. La ricerca mette in evidenza tutte le occorrenze nel capitolo della parola desiderata (Figura 3.5) e mostrerà un alert nel caso di nessun risultato.

The screenshot shows the PhiloEditor 6.0 interface. At the top, there is a navigation bar with 'PhiloEditor 6.0', 'INFO', 'AIUTO', 'STATISTICHE', 'OPERE', and 'CONTRIBUTI'. A search bar contains the word 'mano'. Below the navigation bar, the page title is 'TOMO PRIMO'. The main content area displays the 'INTRODUZIONE' section of a text document. The word 'mano' is highlighted in yellow in the text. On the right side, there is a sidebar with a search filter 'VISTA' and a list of correction methods and categories, including 'Metodologie correttive' and 'Categorie correttive'.

Figura 3.5: Ricerca nel testo.

3.1.2 Limiti di PhiloEditor (edizioni passate)

Uno dei limiti principali di PhiloEditor, molte volte ribadito dalla committenza filologa, è la sua interfaccia.

Questa, in molti casi, si rivela inadeguata all'utilizzo da parte di un neofita o di un utente che non possenga una buona conoscenza del web.

Più nello specifico, l'applicazione risulta poco intuitiva. I fattori che portano a questa constatazione sono vari.

Primo fra tutti la mancanza di una suddivisione netta in sezioni. PhiloEditor mostra unicamente una sezione di home ed una di lettura. Questo porta ad un'approccio troppo diretto, che non permette all'utente né di entrare gradualmente nell'ambiente, né di avere una visione chiara della struttura del sito.

La costante presenza nella navbar di tool come l'analisi delle statistiche e della ricerca nel testo, i quali funzionano solo una volta aperta l'opera, confondono ulteriormente: l'utente potrebbe tentare di attivarle nonostante sia ancora nella home e così otterrebbe risultati erronei o nessun risultato.

Un'altra mancanza individuata è una buona separazione fra le modalità di visualizzazione, è possibile passare fra le varie tipologie, ma senza una vera distinzione logica fra quella che è una semplice visualizzazione di lettura di un'edizione e quelle che invece confrontano le due edizioni.

Va anche aggiunta la mancata segnalazione della modalità "modifica". Infatti un utente che non abbia effettuato il login non potrà nemmeno visualizzare il pulsante per passare alla suddetta modalità, cosicché qualunque utente che non abbia già familiarità con PhiloEditor potrebbe ignorarne l'esistenza e di conseguenza non utilizzarla (Figura 3.6). Da segnalare anche l'impossibilità di ritornare alla sezione home una volta che si è passati alla lettura.

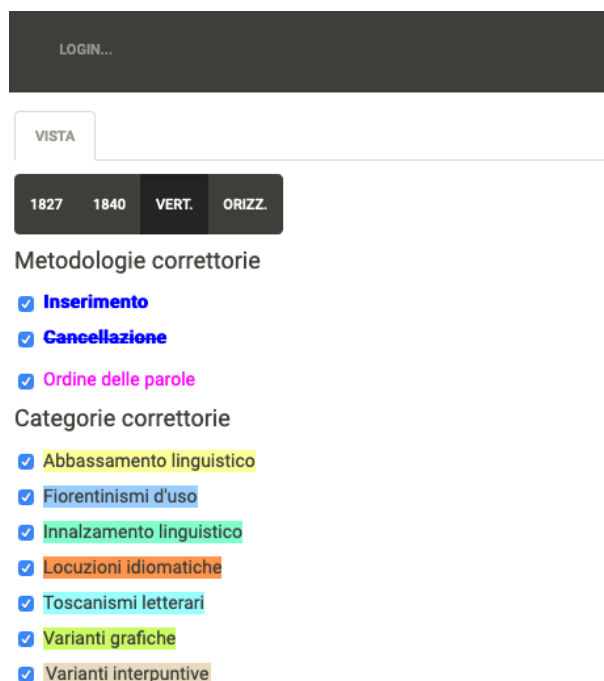


Figura 3.6: Hide del bottone Modifica in caso di mancato login.

Restando nell'ambito delle modalità di visualizzazione delle opere, sono state riscontrate principalmente altre due problematiche.

La prima riguarda la lentezza del caricamento delle opere, in particolare I promessi sposi. La suddivisione in tomi, molto più lunghi rispetto ad un semplice capitolo, richiede un dispendio di tempo non trascurabile nella fase di diffing, oltre ad essere controintuitiva per la maggior parte degli user, i quali imparano a conoscere l'opera nella versione finale del 1840 ("quarantana"), suddivisa in capitoli, e non nella versione del 1827, la "ventisettana" di Fermo e Lucia, suddivisa in tomi.

La seconda problematica invece riguarda la scorretta visualizzazione nella modalità di lettura: quando si desidera leggere una sola delle due edizioni, PhiloEditor mostra le metodologie correttorie basate su un confronto che, agli occhi dell'utente, non dovrebbe avvenire.

Vanno inoltre segnalati alcuni malfunzionamenti nella piattaforma, come il caricamento della pagina molto lungo e a volte impossibile, le statistiche in alcuni casi scorrette ed altri problemi minori di restituzione dei risultati e di visualizzazione, che si presentano con l'utilizzo di browser specifici¹.

Infine il sistema manca di un'appropriata funzione di ricerca fra le opere. PhiloEditor non fornisce all'utente la possibilità di effettuare ricerche di alcun tipo sulle opere presenti.

Una delle principali richieste in merito da parte dei filologi è la possibilità di poter effettuare ricerche basate su parole chiave, le quali restituiscano come risultato i capitoli delle varie edizioni in cui tali parole compaiono e ne fornisca il numero esatto delle occorrenze al loro interno.

¹Ad esempio Firefox, il quale visualizza le versioni de *Le avventure di Pinocchio* invertite.

Capitolo 4

PhiloEditor 7

PhiloEditor è un'applicazione web dedicata all'analisi filologica. Dunque l'obiettivo di PhiloEditor è quello di fornire al moderno filologo uno strumento efficace, che gli permetta di analizzare facilmente e velocemente le opere, confrontarle, marcarle, salvarle e condividerle.

PhiloEditor 7, oggetto di questa dissertazione, propone una serie di strumenti utili a raggiungere questo scopo, tramite un'interfaccia semplice, elegante ed intuitiva che ne facilita e semplifica l'utilizzo. PhiloEditor 7 permette al filologo di non aver necessità di una particolare competenza a livello informatico, ed un'ottima facilità d'accesso a tutte le feature ed i tool di cui l'applicazione dispone.

4.1 Tipi di Utente

Entrando più nel dettaglio si può notare come l'applicativo sia stato pensato principalmente per tre tipi di utente:

1. Utente esperto: colui che già conosce approfonditamente la filologia ed i testi presenti su PhiloEditor e le loro tematiche;
2. Studente universitario: ovvero uno studente che conosce i testi presenti sulla piattaforma in maniera superficiale o non particolarmente

approfondita e che non ha particolare dimestichezza nel campo della filologia;

3. Studente di scuola secondaria di primo e secondo grado: un utente che non conosce o conosce solo in maniera approssimativa i testi e le tematiche trattate e che non ha competenze nella filologia.

Il processo di costruzione del design dell'interfaccia ha seguito le possibili esigenze di ciascun tipo di utenza sopraelencato. Si è dunque giunti al risultato di un'interfaccia semplice ed intuitiva, così da permettere un facile accesso sia ai neofiti che ai più esperti.

Da quanto detto sopra, si possono individuare tre principali tipi di attività all'interno di PhiloEditor, dedicate al tipo di utenza che intenda usufruire della piattaforma:

1. Utente Esperto: studio comparativo tra le edizioni della stessa opera in ottica dell'evoluzione nel tempo della stessa con conseguente possibilità di creare la propria edizione personale per finalità di ricerca;
2. Studente Universitario: studio comparativo tra le edizioni della stessa opera in ottica dell'evoluzione nel tempo della stessa per finalità di apprendimento;
3. Studente di scuola secondaria di primo e secondo grado: semplice lettura o studio di una singola edizione con possibilità di semplici comparazioni esemplificative a scopo didattico.

4.2 Funzionalità del sistema

Nella versione 7 di PhiloEditor si è andati incontro alle richieste della committenza, risolvendo i limiti e le problematiche che inficiavano l'esperienza di utilizzo della piattaforma.

PhiloEditor 7 presenta un'interfaccia totalmente ripensata. L'idea alla base è quella della semplicità di fruizione e la progressiva immersione nell'esperienza, cosicché anche un'utente senza particolari capacità informatiche possa utilizzare l'applicazione senza difficoltà. L'utente viene accolto da un

sito dall'aspetto accattivante e pulito. Questo è stato voluto in accordo con le statistiche aggiornate al 2020 riguardanti il web design [13], in cui viene affermato che:

«...94% of website users first impressions are design related whereas 94% of people cited web design as the reason they mistrusted or rejected a website.» [13]

«It only takes 0.5 seconds to form an opinion about a website.» [13]

Si è perciò optato per un design semplice, squadrato, mobile friendly, e con una palette cromatica sobria ma che ben si adattasse a guidare l'utente all'interno dell'applicazione e ad evidenziare le funzionalità e le sezioni del sito.

Oltre che al miglioramento grafico, sono state implementate nuove soluzioni e correzioni che risolvessero i problemi di rallentamento e malfunzionamento.

4.2.1 Home

La prima pagina a cui si accede è quella della home, questa funge da hub per accedere alle opere disponibili e all'introduzione al sito. La home è suddivisa in tre parti: una schermata di benvenuto (Figura 4.1), una introduttiva (Figura 4.2), da cui è possibile accedere all'introduzione completa, ed una di presentazione delle opere disponibili (Figura 4.3), da cui è possibile accedere alla sezione dedicata alle opere.

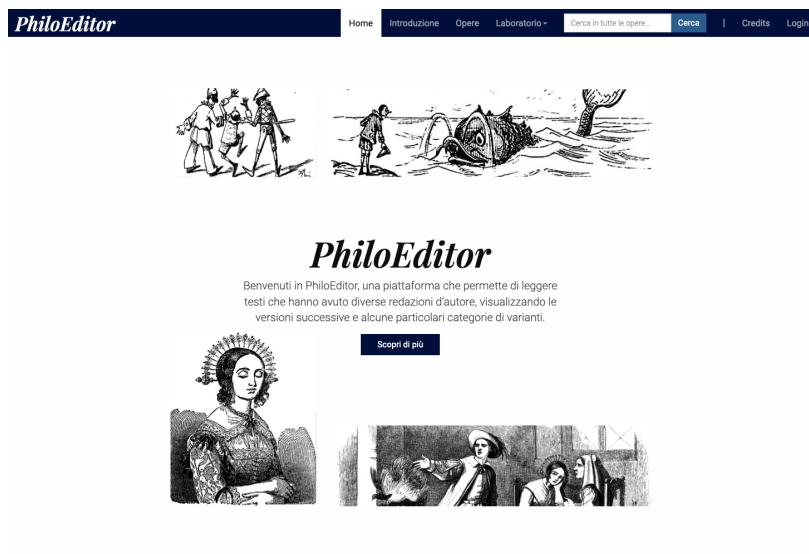


Figura 4.1: Sezione di benvenuto (home).

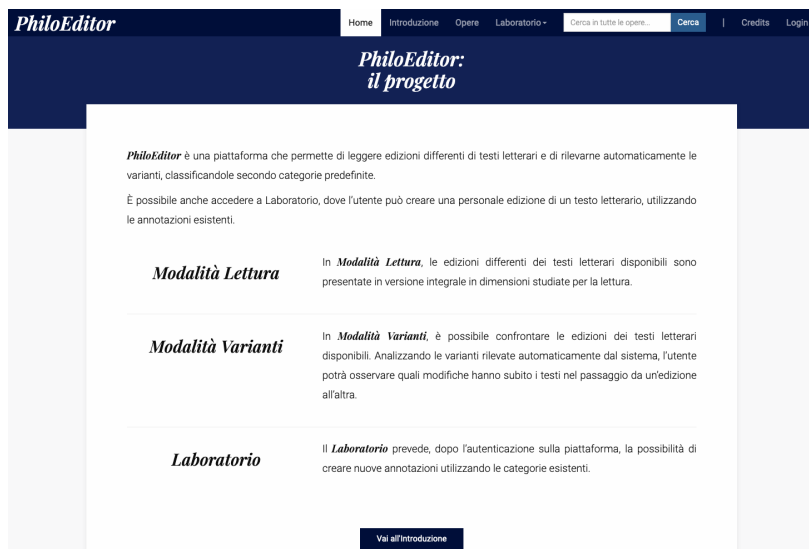


Figura 4.2: Sezione introduttiva (home).

Opere disponibili



I promessi sposi

di Alessandro Manzoni

Edizioni del 1825-1827 (Milano, Vincenzo Ferrario) e del 1840-1842 (Milano, Tipografia Guglielmini e Redaelli).

[Vai a *I promessi sposi*](#)



Le Avventure di Pinocchio

di Carlo Collodi

Edizione del 1881-1883 («Il Giornale per i bambini») ed edizione critica di Castellani Pollidori 1983.

[Vai a *Le Avventure di Pinocchio*](#)

Figura 4.3: Sezione delle opere disponibili (home).

4.2.2 Introduzione

Nella sezione Introduzione, l'utente può visualizzare un'approfondita panoramica del sito.

Qui potrà apprendere tutto ciò che serve riguardo all'utilizzo dell'applicativo, tramite semplici e brevi spiegazioni, oltre che a brevi video tutorial, sulle modalità di visualizzazione delle opere (Figura 4.4).

I colori dei button per passare da un tutorial all'altro richiamano i colori dei button utilizzati per passare tra le varie modalità di lettura, così da facilitare ulteriormente la familiarizzazione con il sito.



The screenshot shows the PhiloEditor website interface. At the top, there is a dark blue navigation bar with the logo 'PhiloEditor' on the left and links for 'Home', 'Introduzione', 'Opere', 'Laboratorio', a search bar with 'Cerca in tutte le opere...' and a 'Cerca' button, and 'Credits' and 'Login' on the right. Below the navigation bar, the main heading is 'Modalità di accesso ai testi'. Underneath, there are three buttons: 'Modalità Lettura' (highlighted in dark blue), 'Modalità Varianti', and 'Laboratorio'. The central content area is titled 'Modalità Lettura' and features a preview of the application's reading interface. This preview shows a dark blue header with 'PhiloEditor' and navigation links. Below the header is a sidebar with an 'Indice' section and a main text area displaying the beginning of 'Le Avventure di Pinocchio' (Chapter II). The text in the preview includes: 'Maestro Ciliegia regalò il pezzo di legno al suo amico Geppetto, il quale lo prese per fabbricare un burattino marionetta, che s'appellò, *fin da allora e fin i suoi giorni*. In quel punto ha bussato alla porta. "Ponete pace," disse il follegname, senza aver la forza di ritrarsi in piedi. Allora entrò in bottega un vecchietto tutto arzilla, il quale aveva nome Geppetto; ma i ragazzi del vicinato, quando lo volevano far montare su tutte le feste, lo chiamavano col soprannome di Polendino, a motivo della sua parrucca gialla, che sceglieva mollemente alla polendina di granturco. Geppetto era bisbetissimo. Così a chiamarlo Polendino! Diventava subito una bestia, e non c'era più verso di tenerlo. "Bene giorno, master! Antano!" disse Geppetto. "Che cosa fate costì per sera?" "Insegno l'alfabeto alle femmine!"

Below the preview, there is explanatory text: 'In **Modalità Lettura**, le edizioni differenti dei testi letterari disponibili sono presentate in versione integrale in dimensioni studiate per la lettura. Dopo aver selezionato una specifica edizione di un'opera da leggere, l'utente potrà scorrere il testo di ciascun capitolo, cambiando eventualmente capitolo utilizzando l'**indice**. L'utente potrà anche effettuare ricerche testuali sull'opera stessa attraverso una **barra di ricerca** e analizzare all'interno del testo in versione integrale l'occorrenza di ciascun risultato.'

Figura 4.4: Sezione di introduzione.

4.2.3 Opere

Accedendo alle Opere si ha una prima sezione dove è possibile scegliere quale opera visualizzare (Figura 4.5). La scelta di realizzare questa sezione di intermezzo è stata voluta per aumentare la chiarezza e per facilitare una futura introduzione di una nuova opera all'interno del sito. Infatti si è optato per un metodo di rappresentazione a "card", le quali contengono un riassunto delle informazioni riguardanti l'opera in questione. Questo permetterà di aggiungere nuove opere velocemente all'interno della struttura, inserendo una nuova card coi dati aggiornati secondo la nuova opera.

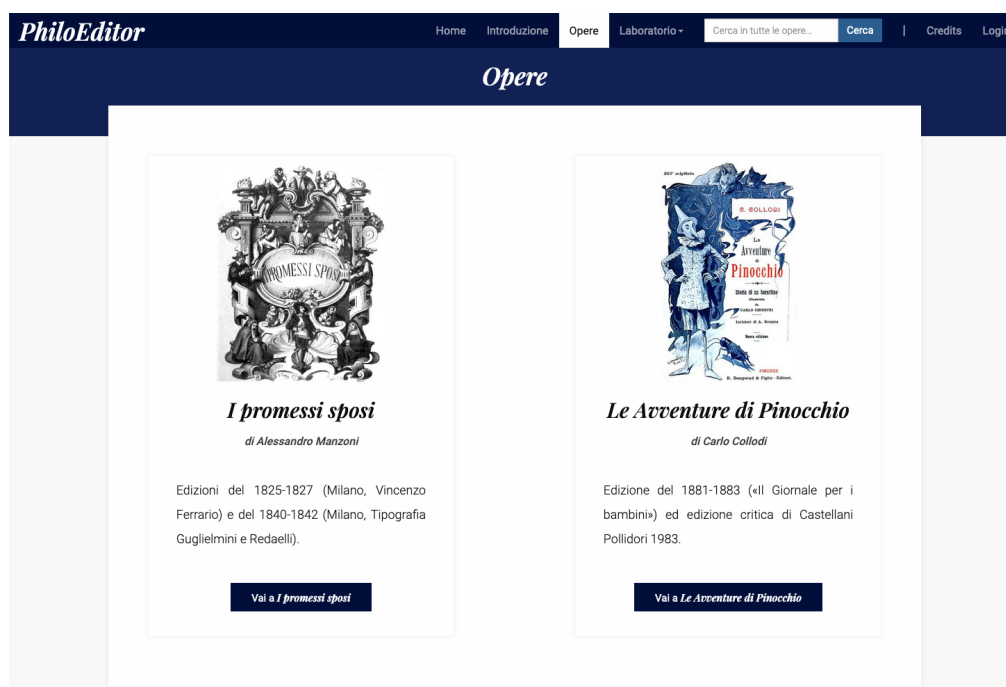


Figura 4.5: Sezione delle opere.

Ad ogni opera è dedicata una sezione specifica, in cui è possibile scegliere il tipo di modalità di visualizzazione (Figura 4.6). Anche in questo caso è stata scelta una suddivisione per card. La creazione

di questa sezione ha permesso la rimozione del menu ad albero presente nella navbar dedicato alle opere, in favore di un menu lineare dedicato a ciascuna opera. Questo permette all'utente di avere una visione chiara e pulita di ciò che andrà a leggere e in quale modalità di lettura, che verranno esposte in seguito.

Da aggiungere che nel caso de *I promessi sposi* si è optato per una suddivisione dell'opera per capitoli e non per tomi. Questo ha migliorato notevolmente i tempi di attesa oltre che a fornire maggiore coerenza.

The screenshot displays the 'Edizioni disponibili' (Available Editions) section for the work 'I promessi sposi'. The page features a dark blue header with the 'PhiloEditor' logo and navigation links: Home, Introduzione, Opere, Laboratorio, and a search bar. Below the header, the title 'I promessi sposi' is centered, with a 'Torna indietro' link on the left. The main content area is divided into two columns, each presenting an edition:

- Left Column:** Features the cover of the first edition (1825-1827). Below the cover, the title 'I promessi sposi (1825-1827)' is followed by a paragraph: 'La prima edizione de *I promessi sposi* viene pubblicata a puntate, in tre tomi, presso l'editore Ferrario di Milano tra il 1825 e il 1827. Il modello linguistico di riferimento è il toscano scritto, come poteva essere reperito nei vocabolari e nei testi di scrittori toscani. È il risultato delle revisioni iniziate già all'altezza della stesura del primo manoscritto, il *Fermo e Lucia*.' A 'Leggi I promessi sposi (1825-1827)' button is at the bottom.
- Right Column:** Features the cover of the second edition (1840-1842). Below the cover, the title 'I promessi sposi (1840-1842)' is followed by a paragraph: 'Pubblicata a Milano presso la tipografia Guglielmini Redaelli a puntate tra il 1840 e il 1842, la seconda stampa de *I promessi sposi* è frutto della svolta linguistica maturata durante il soggiorno di Manzoni a Firenze nell'estate del 1827, e orientata al fiorentino parlato. L'edizione, oltre alla revisione linguistica, si caratterizza per le litografie ad opera di Francesco Gonin.' A 'Leggi I promessi sposi (1840-1842)' button is at the bottom.

At the bottom center, a red button reads 'Confronta le due edizioni de *I promessi sposi*'.

Figura 4.6: Sezione di un'opera specifica.

4.2.4 Modalità di visualizzazione

Una volta selezionata la modalità di visualizzazione dell'opera e il capitolo desiderato, PhiloEditor caricherà il capitolo in questione, mostrandolo all'utente.

Le modalità possibili sono tre: Lettura, Varianti e Laboratorio.

La modalità di visualizzazione Varianti è la modalità su cui PhiloEditor è incentrato. In questa modalità l'utente può leggere il capitolo selezionato mettendo a confronto le due edizioni (Figura 4.7). La modalità Varianti è principalmente dedicata all'utente di tipo 1 e 2¹.

The screenshot shows the PhiloEditor interface in the 'Modalità Varianti' view. The page title is 'I promessi sposi' and the chapter is 'Capitolo I'. The main text area displays the beginning of the chapter with various words in different colors and sizes, indicating differences between editions. On the left side, there is a sidebar with navigation options like 'Indice', 'Metadati dell'edizione', 'Varianti', and 'Metodologie correttive'. The top navigation bar includes 'Home', 'Introduzione', 'Opere', 'Laboratorio', and search options.

Figura 4.7: Modalità Varianti.

In questa sezione il menu offre varie opzioni: scegliere una visualizzazione di confronto orizzontale o verticale, navigare l'indice dei capitoli dell'opera selezionata, effettuare ricerche nel testo dell'opera, visualizzare i metadati, una legenda delle metodologie correttive e la possibilità di mostrarle o nasconderele nel testo a seconda delle esigenze.

¹Facendo riferimento ai tipi di utenza descritti nel paragrafo 3.1.

Si è quindi optato per una divisione logica e netta del menu, che offra un accesso semplice a tutte le opzioni dedicate a questa modalità e solamente ad esse per evitare fraintendimenti.

Si è anche scelto di rimuovere in questa sezione la parte delle categorie correttive, in quanto non presenti in questa opzione di visualizzazione.

Come si può notare è possibile, tramite i bottoni in alto, passare da una modalità di visualizzazione all'altra facilmente.

La modalità Lettura (Figura 4.8) è una modalità di visualizzazione totalmente inedita, dedicata alla lettura di un'unica edizione e ad un'utenza di tipo 2 e 3.



Figura 4.8: Modalità Lettura.

Tramite un piccolo menù a scomparsa si può scegliere quale delle due edizioni leggere, navigare fra i capitoli, visualizzare i metadati ed effettuare ricerche di parole nel testo.

Questa soluzione è stata adottata per facilitare all'utente la separazione logica fra lettura e confronto delle edizioni. Rispetto alle versioni precedenti

di PhiloEditor, PhiloEditor 7 risolve anche i problemi legati all'erronea presenza delle metodologie correttorie in queste modalità: ora queste metodologie correttorie non vengono più mostrate, favorendo la semplice lettura.

Ultima modalità di visualizzazione è quella di Laboratorio.

Nel Laboratorio l'utente, previa autenticazione, potrà effettuare le sue personali marcature secondo le categorie correttorie specifiche per l'opera fornite da PhiloEditor (Figura 4.9). Per effettuare una marcatura basterà evidenziare le parole individuate e selezionare la corrispondente categoria correttoria: le parole si evidenzieranno del colore della categoria e si verrà a creare una marcatura.



Figura 4.9: Modalità Laboratorio.

Il bottone per passare a questa modalità è sempre visibile anche in caso di utente non autenticato, tuttavia al suddetto utente, nel caso di tentativo d'accesso a tale modalità, sarà notificata la mancata autenticazione e non gli verrà consentito di visualizzare questa modalità.

Per maggior chiarezza, sono stati inseriti nel menu laterale i bottoni di salvataggio e nuovo salvataggio, a differenza delle vecchie edizioni dell'applicazione, in cui erano posti nel menu dell'utente.

Questa modalità eredita il menu laterale della modalità varianti, ma è anche possibile, oltre che effettuare le marcature, salvare il proprio testo marcato, scegliere se visualizzare o nascondere metodologie e categorie e visualizzare le statistiche riguardanti la propria marcatura.

Questa modalità di visualizzazione è principalmente dedicata ad un'utenza di tipo 1, ma può essere utilizzata anche da un'utente di tipo 2 per esercitarsi ed iniziare a prendere familiarità con l'analisi critica dell'opera.

4.2.5 Laboratorio

Come si può già evincere dalle precedenti Figure, costante è la presenza della navbar. Anch'essa è stata ripensata in virtù di una maggiore fruibilità e coerenza coi contenuti.

Oltre alla possibilità di accedere alle sezioni precedentemente viste sono presenti altre opzioni.

La prima che si incontra è il menù ad albero del Laboratorio, qua è possibile accedere sia alle proprie marcature precedentemente salvate sia a quelle effettuate da altri. Il menu è suddiviso per scuole, suddivise per utenti. Al momento sono presenti dei laboratori esemplificativi che permettono ad i nuovi utenti di avere un riferimento per le loro attività.

La pagina visualizzata selezionando un capitolo dal suddetto menu è molto simile a quella precedentemente vista delle modalità di visualizzazione. Da notare l'assenza del bottone "modalità Lettura", in quanto, in questo caso, la pagina è votata al confronto e all'attività di marcatura. Le modalità di visualizzazione sono dunque due: Visualizza e Laboratorio.

Nella modalità Visualizza (Figura 4.10) si ha una visualizzazione del testo molto simile alla modalità Varianti precedentemente esposta. Unica variante è la presenza delle categorie correttive nel menu, in quanto, in questo caso, presenti nel testo. Inoltre possono essere visualizzate correttamente le statistiche, rappresentate sia numericamente sia con diagrammi a torta.

Da notare inoltre il button “Esporta TEI”, il quale permette all’utente di salvare localmente la propria copia del capitolo marcato.

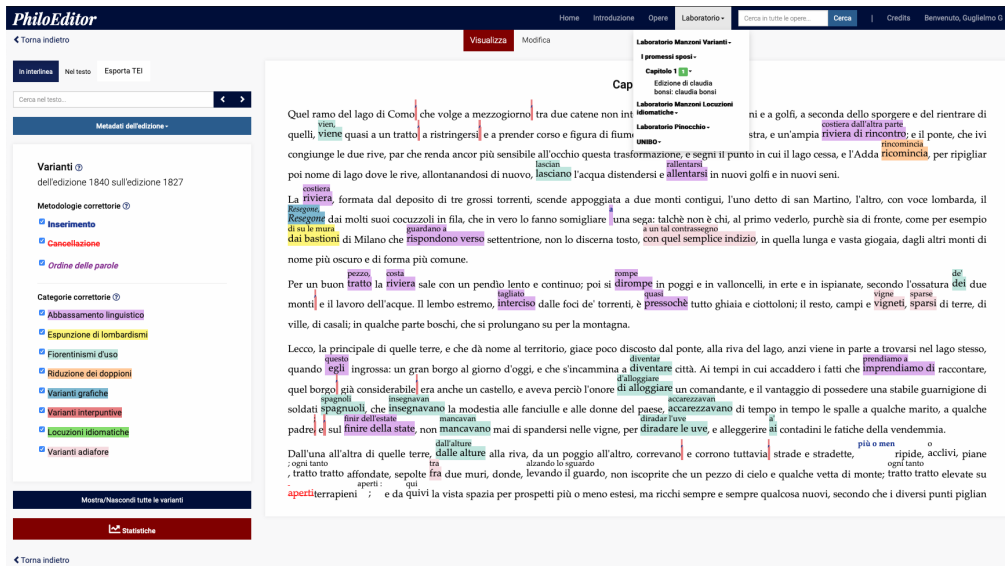


Figura 4.10: Visualizza.

La seconda modalità, Modifica (Figura 4.11), è simile alla modalità Laboratorio vista precedentemente.

In questo caso però non verrà visualizzato un capitolo “pulito”, ma sarà possibile effettuare nuove marcature su un capitolo precedentemente marcato, salvarlo come proprio e/o salvarlo localmente tramite il button “Esporta TEI”.

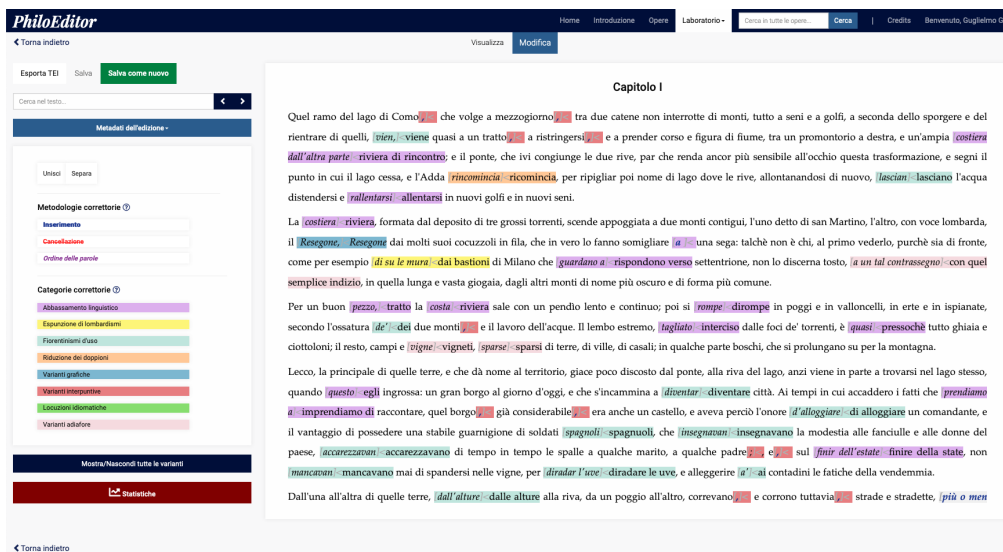


Figura 4.11: Modifica.

4.2.6 Ricerca sulle opere

Infine la navbar presenta una nuova barra di ricerca. Questa barra permette di visualizzare una pagina in cui vengono restituite tutte le occorrenze della parola, o parole², chiave desiderata all'interno di tutti i capitoli di tutte le opere (Figura 4.12).

I risultati vengono suddivisi in card, contenenti il titolo dell'opera, il capitolo e l'edizione. Inoltre è presente un button "Leggi", che permette di leggere il capitolo in questione, oltre al button "Mostra le occorrenze". Selezionando quest'ultimo si aprirà un accordion contenente tutte le occorrenze della parola ricercata inserite in una porzione di contesto testuale. Selezionando una di queste occorrenze sarà possibile leggere il testo partendo all'altezza dell'occorrenza desiderata (Figura 4.13).

²Nel caso di più parole chiave, la ricerca restituirà esclusivamente i capitoli in cui tutte le parole sono presenti contemporaneamente.

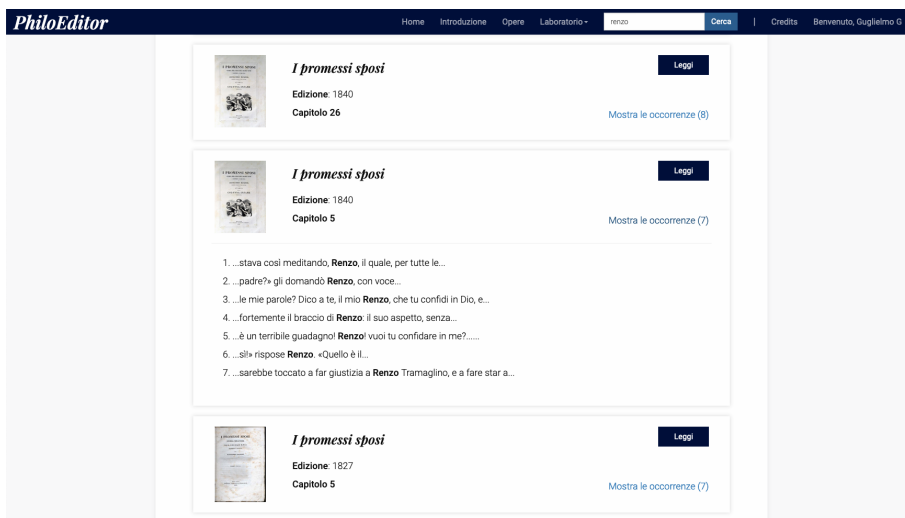


Figura 4.12: Sezione dei risultati di ricerca sulle opere.

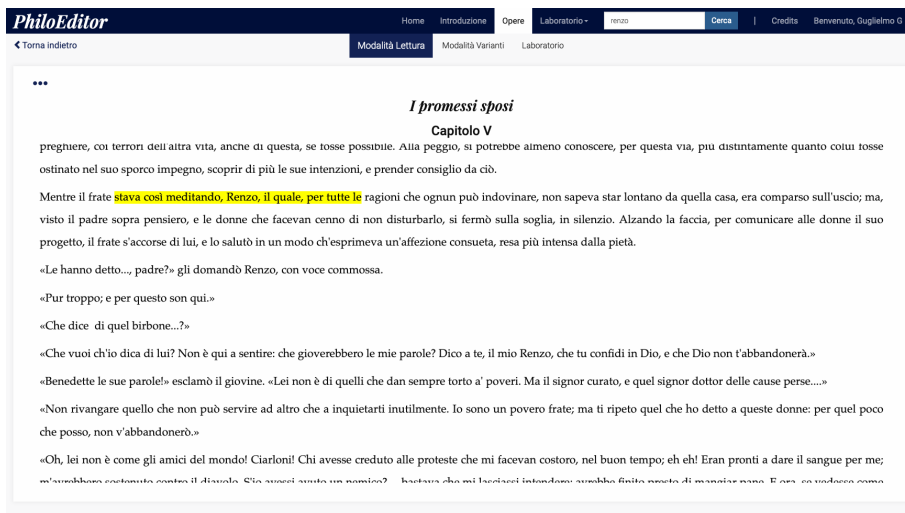


Figura 4.13: Visualizzazione dell'occorrenza selezionata.

Questa pagina è lasciata intenzionalmente bianca.

Capitolo 5

Nuova struttura: PhiloEditor 7

La settima versione di PhiloEditor è stata totalmente ripensata a livello di User Experience / User Interface, oltre a presentare aggiornamenti alle passate funzionalità ed alcune aggiunte inedite.

5.1 Tecnologie

Per realizzare PhiloEditor 7 si è fatto uso di svariate tecnologie. Alcune di queste sono già note e vengono qua semplicemente elencate:

- HTML, come linguaggio di markup;
- CSS, per gli stili;
- Bootstrap, per migliorare il design e rendere la piattaforma mobile-friendly;
- JavaScript, come linguaggio per il front-end;
- jQuery, per facilitare e velocizzare l'utilizzo di JavaScript;
- PHP, per la parte di back-end.

Oltre a queste tecnologie, sono stati utilizzati due ulteriori strumenti: TNTSearch e wikiEd diff.

5.1.1 TNTSearch

«TNTSearch is a full-featured full text search (FTS) engine written entirely in PHP.» [14]

«... [TNTSearch] is extremely fast and since the package is using state of the art relevance algorithms the relevance is much higher than in other PHP solutions. The performance of the package is great because the index is built on top of a sqlite database which is an extension written in C and comes default with PHP. And you cannot beat C...» [15]

TNTSearch nasce come un potente pacchetto di ricerca testuale, creato da TNTStudio.

Il pacchetto è molto versatile ed attualmente permette di effettuare ricerche estremamente rapide non solo nei testi (un esempio dei tempi nella Tabella 7.1), ma anche nelle mappe e, secondo il team, a breve anche all'interno di videogiochi PS4 [14].

Il pacchetto fornisce ulteriori opzioni, quali, ad esempio, l'highlight delle parole chiave individuate e una classificazione testuale.

5.1.2 wikEd diff

«wikEd diff is a free JavaScript visual diff library for inline text comparisons. It is the only available JavaScript diff library that detects and highlights block moves and that works on the word and character level. While wikEd diff has been developed and optimized for comparing Wikipedia source texts, it works great for any type of text, including program code. The library is customizable, has Unicode and multilingual support...» [10]

wikEd diff è una libreria JavaScript potente e versatile, creata per effettuare il diffing sui testi. La libreria ha inoltre il grande vantaggio di essere completamente modificabile a piacimento, così da poterla adattare alle esigenze dell'utilizzatore.

5.2 Front-end

Vengono qui descritte le componenti front-end, ovvero tutte le funzionalità che si attivano clint-side.

Per queste funzionalità è stato fatto largo uso di Javascript e il framework jQuery, oltre che della libreria wikiEd diff per effettuare l'operazione di diffing. Sono state inoltre utilizzate alcune componenti di Bootstrap.

5.2.1 Caricamento iniziale

Quando un'utente si connette al sito vengono caricate tutte le funzionalità ed i riferimenti che si andranno a connettere ai vari elementi del DOM¹. Vengono inoltre nascoste le porzioni di sito non accessibili al caricamento iniziale.

5.2.2 Navigazione

All'interno del sito è ora possibile navigare tra le varie sezioni, è stata aggiunta inoltre la possibilità di tornare alle schermate precedenti.

Sono presenti anche delle funzionalità per proporzionare ed adattare le dimensioni di alcuni elementi a seconda dello schermo utilizzato per visualizzare il sito.

5.2.3 Diffing

Il sistema di diffing fornito dalla libreria wikiEd diff permette un confronto word-based che viene mostrato tramite la visualizzazione dell'edizione meno recente a cui vengono sovrapposte le varianti dell'edizione più recente.

Quando viene selezionato un capitolo da visualizzare, viene chiamata la funzione di diffing, la quale mostra una barra di caricamento, per poi presentare il risultato. Per migliorare le performance si è optato per una suddivisione delle opere esclusivamente in capitoli, abbandonando i tomi, i

¹In alcuni casi la versione 6.5 tendeva a bloccarsi, il problema è stato individuato all'interno di una di queste funzioni di caricamento iniziale. La funzione è stata rimossa ed è stata sostituita con un'altra più stabile.

quali richiedevano tempi di caricamento non indifferenti.

La funzione è suddivisa in vari step, il completamento di tali passaggi determina la percentuale di completamento mostrata dalla barra di caricamento. Inizialmente vengono forniti i testi delle due edizioni, ricevuti dal server. Viene poi verificato se la richiesta di visualizzazione comprende anche l'individuazione di un'occorrenza specifica nel testo, se sì, viene individuata l'occorrenza in questione, altrimenti questo passaggio viene ignorato.

In seguito viene chiamata la funzione di diffing, e, fatto ciò, il risultato viene parsato in modo da ottenere una porzione di codice html, in cui le varianti vengono identificate mediante il tag ``, che viene appesa al box contenente il testo. Viene inoltre reso compatibile il documento con l'esportazione TEI. Nel caso fosse presente la richiesta di individuazione di una specifica occorrenza, dopo aver effettuato l'append, viene fatto scrollare il testo fino all'altezza dell'occorrenza desiderata, altrimenti anche questo passaggio viene ignorato.

Si è inoltre optato per ritardare il più possibile la visualizzazione del testo risultante per evitare sgradevoli cambiamenti repentini dovuti al fixing delle dimensioni degli elementi del DOM.

5.2.4 Modalità di visualizzazione

Una volta conclusa la fase di diffing, viene mostrato il documento nella modalità richiesta dall'utente.

Mentre per le due modalità di visualizzazione Varianti e Laboratorio si è mantenuta una struttura simile alle precedenti edizioni², la Modalità Lettura è stata costruita ex novo. Partendo dalla base data dalla modalità confronto, è stata creata una funzione che la modificasse permettendo la lettura di una singola edizione del capitolo desiderato.

Per facilitare ulteriormente la navigazione fra i capitoli della stessa opera è stato aggiunta una funzionalità in grado di fornire, anche durante la lettura, l'indice dei capitoli dell'opera in questione.

²Rispettivamente le precedenti modalità erano Vista e Modifica.

5.2.5 Ricerca nel testo

Inserendo testo nell'apposita barra di ricerca, le corrispondenze nel testo vengono messe in risalto tramite un highlight. Ogni volta che viene inserito un nuovo carattere la ricerca viene nuovamente effettuata. Premendo il bottone di avanzamento è possibile passare da un'occorrenza all'altra. In caso di mancanza di occorrenze viene mostrato un error.

5.2.6 Statistiche

Dopo aver effettuato una chiamata ajax al server, viene effettuato un confronto tra gli id delle statistiche ottenute e quelli dei documenti.

Ricevute così le informazioni, viene poi costruita la tabella contenente quest'ultime ed infine viene mostrata all'utente sotto forma di modal³.

5.2.7 Caricamento dei documenti

La richiesta ajax per ottenere le informazioni riguardo ai documenti è suddivisa in due parti: una per i testi dei capitoli ed una per i documenti marcati.

Ottenute le informazioni, i menu vengono resettati e tali informazioni inserite, cosicché, in caso di cambiamenti, i menu restino sempre aggiornati.

5.2.8 Salvataggio delle marcature

Nel salvataggio dei documenti inizialmente ci si occupa di prelevare le informazioni necessarie per il documento TEI, in seguito viene effettuata una chiamata di tipo ajax per aggiornare o creare il documento e le rispettive informazioni.

Mentre alcune informazioni vengono prelevate automaticamente, altre dovranno essere inserite manualmente dall'utente, i campi richiesti sono: Nome autore dell'intervento, Responsabilità dell'autore, Luogo di pubblicazione, Licenza per il riutilizzo dei dati e Interventi nell'edizione digitale.

³Anche per questa funzione è stato necessario effettuare una serie di bug fix, in quanto i risultati numerici risultavano scorretti.

5.2.9 Esportazione TEI

La funzionalità di esportazione del TEI non è mutata rispetto alle passate versioni. Tale funzione controlla le informazioni ed il testo, suddivide il documento basandosi sulle porzioni contenenti la classe `replace` o meno, per poi esportare il TEI.

5.2.10 Ricerca nelle opere

La funzionalità di ricerca nelle opere permette di effettuare ricerche di parole chiave su tutti i capitoli di tutte le opere.

Inizialmente l'utente dovrà inserire la parola (parole) chiave⁴.

Dopo aver effettuato una chiamata ajax al server, la funzione verifica che tutti i risultati siano coerenti, filtrando le occorrenze ed i risultati non pertinenti. Una volta fatto ciò, confronta coi dati delle opere i risultati e riorganizza le informazioni ricevute, inserendole in un array di JSON.

Completato questo passaggio, viene costruito l'html per mostrare all'utente i risultati, suddivisi in card.

Prima che l'html sia realizzato, una funzione analizza e sistema i possibili errori ed eccezioni all'interno delle occorrenze, in modo da renderle individuabili all'interno del testo. Si rende necessario questo passaggio in quanto i testi salvati sul server, su cui viene effettuata la ricerca, non corrispondono all'output fornito dal diffing, dunque le occorrenze non verrebbero poi individuate.

Una volta costruite le card, sarà da queste possibile leggere il capitolo trovato, e in tal caso verrà semplicemente invocata la "classica" funzione di diffing, altrimenti sarà possibile selezionare una specifica occorrenza. Se quest'ultima opzione viene selezionata, la funzione individuerà nel testo del capitolo individuato l'occorrenza, alla quale assegnerà un identificativo nell'html.

⁴Si è preferito evitare di mostrare risultati all'inserimento di ogni carattere (come nella ricerca nel testo, paragrafo 4.2.5), in quanto si necessiterebbe di effettuare una continua ricerca che avverrebbe su scala troppo vasta. La ricerca viene infatti effettuata su ogni parola di ogni testo di ogni edizione di ogni capitolo di ogni opera, e i risultati, specialmente quando vengono inseriti pochi caratteri, rallenterebbero troppo l'applicazione.

Una volta finita l'operazione di diffing, il testo verrà fatto scrollare all'altezza dell'occorrenza scelta, la quale sarà evidenziata tramite un background colorato mediante CSS. Dopo un tempo prestabilito, il background viene rimosso. Questo tempo è stato calcolato per permettere all'utente di visualizzare ciò che ha richiesto, evitando allo stesso tempo che tale background venga poi salvato in caso di marcatura del documento.

```
{
  titolo : 'I promessi sposi',
  edizione : '1840',
  capitolo : '24',
  libro : 1,
  pathVersione1: 'sources/01 - I promessi sposi/25 -
    Capitolo 24/49.txt',
  ed1: '1827',
  pathVersione2: 'sources/01 - I promessi sposi/25 -
    Capitolo 24/50.txt',
  ed2: '1840',
  pos: 24,
  modalitaLettura: 'secondaVersione',
  occorrenze: [
    '...agitarlo troppo. La lontananza di <b>Renzo</b>,
      senza nessuna...',
    '...trovar la maniera di far che <b>Renzo</b> si
      rassegnasse anche lui,...',
    ' E <b>Renzo</b>? ; disse...',
    '...poco che sapeva della storia di <b>Renzo</b>.'
  ]
};
```

Listing 5.1: Esempio di un output di ricerca.

5.2.11 Sicurezza

L'applicazione presenta inoltre una funzione di crittografia, che permette di criptare le password degli utenti, le quali verranno poi salvate sul server.

In questo modo, anche se il file JSON contenete le password fosse acceduto, non sarebbe possibile risalire alle password in plain text.

5.3 Back-end

In questa sezione verranno analizzate le componenti che vanno a formare il back-end. Per realizzarle principalmente è stato utilizzato il linguaggio PHP.

5.3.1 Richieste sui documenti

Come accennato nel paragrafo 4.2.7, la richiesta è stata suddivisa in due parti. Una richiesta viene fatta al file `getFilesS.php`, per ricevere i dati riguardanti le opere presenti sul server, che risiedono nella cartella `sources`, mentre la seconda viene fatta al file `getFilesC.php`, per ottenere i dati riguardanti i capitoli marcati, residenti nella cartella `contributors`.

Il codice dei due file è simile, la differenza sta che nel secondo vengono aggiunte informazioni aggiuntive riguardanti la marcatura del documento, quali autore, data, ecc. Le due cartelle da cui vengono prelevate le informazioni sono strutturate in maniera gerarchica ed ordinata.

Nel caso della chiamata a `getFilesS.php`, esso andrà a ricercare le informazioni nella cartella `sources`, in cui si ha questa struttura:

```
sources
-> 01 - I promessi sposi
      -> 00 - Metadata
          01 - Introduzione
              -> 01.txt
              02.txt
          02 - Capitolo 1
              -> 03.txt
              04.txt
          03 - Capitolo 2
          ...
```

```
02 - Le avventure di Pinocchio
-> ...
```

Listing 5.2: Struttura della cartella sources.

Nella sottocartella Metadata è presente il file `style.json`, contenente le informazioni relative alle possibili categorie correttive presenti nell'opera. Da questo file vengono prelevate le informazioni riguardanti i nomi e gli stili CSS di ciascuna categoria.

Effettuando invece una chiamata a `getFilesC.php`, questo andrà a prelevare le informazioni dalla cartella contributors, così strutturata:

```
contributors
-> ...
  04 - UNIBO
    -> 01 - I promessi sposi
      -> 00 - Metadata
        01 - Introduzione
          -> 1.html
          ...
    02 - Le avventure di Pinocchio
      -> ...
```

Listing 5.3: Struttura della cartella contributors.

In questo caso la sottocartella Metadata contiene le informazioni riguardo alla marcatura, nel file `infoDoc.json`, come il creatore, la data di creazione, ecc. e `stats.json`, contenente le statistiche.

In entrambi i casi la ricerca viene effettuata in maniera ricorsiva, abbinando le informazioni coi relativi documenti all'interno di un file JSON che viene poi consegnato al client.

5.3.2 Ricerca nelle opere

Per effettuare la ricerca sulle opere mediante una parola (o parole) chiave viene effettuata una chiamata al file `search.php`, il quale restituisce un

array in JSON contenente i dati necessari all'applicazione per mostrare correttamente i risultati di tale ricerca.

Per raggiungere questo obiettivo è stato utilizzato il pacchetto PHP TNTSearch. Il pacchetto, una volta installato ha necessitato della creazione di un file di tipo `.index`, il quale indicizza tutti i contenuti di una directory selezionata, in questo caso, piuttosto prevedibilmente, la scelta è ricaduta sulla cartella `sources` (mostrata nel paragrafo 4.3.1).

Una volta generato il file `.index` è stata creata la vera funzione di ricerca. Quando la funzione riceve le parole chiave, questa effettua le ricerche sull'indice, raccoglie tutti i path dei file che abbiano tali parole al loro interno e tutte le occorrenze di quelle parole, e infine restituisce i risultati (Listing 5.4).

TNTSearch tuttavia non nasce per ricerche su opere letterarie, è stato dunque necessario apportare una serie di modifiche per fare in modo che i risultati fossero coerenti. Il primo passo è stato quello di dotare la funzionalità della capacità di individuare tutte le occorrenze all'interno di un testo, e che queste fossero coerenti e con le opportune correzioni, in modo che poi, il lato client-side, fosse capace di mostrarle correttamente.

Inoltre è stata applicata la funzione interna al pacchetto "highlight", la quale permette di porre un tag html sulla parola chiave in modo che poi possa venire evidenziata in fase di visualizzazione da parte dell'utente. Anche in questo caso sono state applicate delle modifiche in modo che la parola venisse posta all'interno del tag ``.

```

{
  'id' : '50',
  'path' : '\\var\\www\\html\\phed6.6.1\\sources\\01 - I
    promessi sposi\\25 - Capitolo 24\\50.txt',
  'occorrenze' : [
    '...agitarlo troppo. La lontananza di Renzo</b>,
      senza nessuna...',
    '...trovar la maniera di far che Renzo</b> si
      rassegnasse anche lui,...',
    '...che aveva un certo nome!@ E Renzo</b>? disse
      ...',
    '...poco che sapeva della storia di Renzo</b>.@ Ho
      sentito parlare...'
  ]
}

```

Listing 5.4: Esempio di un risultato per la ricerca effettuata sulla parola “renzo” con le rispettive occorrenze.

5.3.3 Statistiche

Questa funzionalità permette di visualizzare le statistiche riguardanti le marcature.

La funzione si limita a prelevare e consegnare i dati appartenenti al file stats.json andandolo a scegliere nella giusta sottocartella Metadata.

5.3.4 Salvataggio delle marcature

Una volta marcato un documento è possibile salvarlo come nuovo o aggiornarlo (salvarlo) nel caso sia pre esistente.

Come è possibile notare nel paragrafo 4.3.1, la cartella contributors è suddivisa nelle sottocartelle appartenenti alle scuole. Se l’utente sta richiedendo di aggiornare un documento, semplicemente il documento verrà individuato ed aggiornato.

Se invece l’utente tenterà di salvare una nuova marcatura, allora tale documento verrà posto all’interno della cartella della rispettiva opera. Mentre

alcune informazioni vengono inserite automaticamente, altre dovranno essere inserite dall'utente al momento della richiesta di salvataggio.

Nel caso che l'utente non appartenga a nessuna delle scuole, ovvero il nome della sua scuola non faccia match con nessuna delle sottocartelle esistenti, ne viene creata una nuova col nome della scuola di appartenenza, seguita dalla creazione della sottocartella dedicata all'opera, seguita a sua volta dall'inserimento di una sottocartella del capitolo, in cui verrà poi posto il file.

Estremamente simile il caso in cui l'utente non abbia mai salvato marcature fatte su una specifica opera e/o di un capitolo, ma già è presente una sottocartella che faccia match col suo nome della scuola. In tal caso si provvederà ad aggiungere una sottocartella col nome dell'opera e/o del capitolo all'interno della cartella della scuola, e successivamente verrà salvato il nuovo file.

Nel caso di mancata riuscita del salvataggio, la situazione di impossibilità viene notificata tramite error⁵.

5.3.5 Condivisione delle marcature

Questa funzionalità implementa la possibilità di condividere con altri utenti le proprie marcature.

La funzione applica un match ed individua utente e proprietario del documento. Fatto ciò inserisce il nome dell'utente scelto nelle informazioni relative al documento.

5.3.6 Registrazione

Il sito dà la possibilità all'utente di registrarsi. Compiuta quest'azione, l'utente potrà accedere ad ulteriori funzionalità fino a quel momento bloccate, quali la possibilità di effettuare le proprie marcature, salvarle e condividerle. Il server riceverà i dati di registrazione inseriti nell'apposito form da parte dell'utente. Una volta ricevuti tali dati, questi verranno salvati temporaneamente.

⁵Anche in questo caso si sono riscontrati alcuni problemi, risolti poi grazie al dialogo con il laboratorio DISI.

Se i dati non faranno match con nessuno degli utenti già presenti, il sistema registrerà la nuova utenza. Altrimenti la richiesta verrà rifiutata.

5.3.7 Login / logout

PhiloEditor permette ad un utente già registrato di effettuare il login e il logout.

In maniera molto simile a quella descritta nel paragrafo 4.3.6, la funzionalità di login effettua un confronto con i dati inseriti dall'utente e quelli presenti sul server degli utenti registrati. Se il match ha successo allora l'utente avrà effettuato il login, altrimenti gli verrà notificata l'impossibilità ad accedere. Per quanto riguarda il logout viene effettuato l'unset e poi i cookie vengono ripristinati.

5.3.8 Aggiornamento della password

Questa funzione permette di aggiornare la propria password.

Viene richiesto l'inserimento della password precedente e della nuova password (due volte). Se tutti i dati forniti fanno match, allora la password viene modificata e da quel momento in poi l'utente potrà utilizzare la nuova password.

Questa pagina è lasciata intenzionalmente bianca.

Capitolo 6

Metodologia di sviluppo software

In questo capitolo viene descritta la metodologia di programmazione utilizzata per realizzare PhiloEditor 7.

Per la realizzazione di questo progetto si sono presentate due principali necessità: uno stretto dialogo con la committenza filologa e la possibilità di fornire continui risultati, anche intermedi, in breve tempo, dato che nel periodo di sviluppo del software i committenti avrebbero effettuato alcune presentazioni al pubblico.

6.1 Extreme Programming

Dopo un primo dialogo con la committenza riguardo al futuro di PhiloEditor e aver considerato le dimensioni del team, di sole due persone, la scelta del metodo di programmazione è ricaduto sul metodo agile, in particolare sull'Extreme Programming (XP).

«Extreme Programming (XP) is an agile software development framework that aims to produce higher quality software, and higher quality of life for the development team.» [16]

Tale metodo è basato su cinque principi fondamentali: comunicazione, feedback, semplicità, coraggio e rispetto [17]. Più nel dettaglio:

1. Comunicazione: l'XP dà grande importanza ad un'appropriata comunicazione, specialmente a quella faccia a faccia.
2. Feedback: attraverso un costante feedback il team può identificare aree critiche e migliorare velocemente il proprio codice.
3. Semplicità: questo concetto è racchiuso nella frase «*what is the simplest thing that will work?* » [17]. Questo implica l'obiettivo di mantenere il codice più semplice e pulito possibile, così da renderne facile la produzione, la comprensione e la manutenzione.
4. Coraggio: Kent Beck ha definito il coraggio come «*effective action in the face of fear*» [18]. Questa definizione mostra una preferenza per l'agire basandosi sugli altri principi, cosicché i risultati non siano dannosi per il team. Serve dunque coraggio per: superare metodi organizzativi che rallentano l'efficacia del team, abbandonare una via che non sta funzionando per imboccarne un'altra, accettare un feedback e agire di conseguenza.
5. Rispetto: i membri del team devono rispettarsi, in modo da comunicare tra loro. Questo faciliterà lo sviluppo del software trovando soluzioni semplici ed efficaci.

Applicando questi principi, sono poi state adottate alcune pratiche tipiche dell'XP per sviluppare e completare PhiloEditor 7. In particolare il Pair Programming.

«Pair Programming means all production software is developed by two people sitting at the same machine.» [19]

Il Pair Programming dunque prevede di abbassare il numero di macchine ma di raddoppiare il numero di programmatori su di esse, così da poter fornire una doppia cura nella produzione del codice, conseguentemente aumentandone la qualità.

Grazie all'applicazione del Pair Programming è stato infatti possibile sviluppare il software velocemente, effettuare bug fixing più facilmente e soprattutto evitare la maggior parte degli errori più banali, come quelli di

battitura, che molte volte richiedono una non indifferente quantità di tempo per essere individuati e risolti.

Prima di applicare il Pair Programming è stato applicato il metodo del Sit Together, ovvero brevi discussioni preliminari sul programma di ogni singola sessione di produzione del software. È stato così possibile svolgere interventi solidi e mirati.

Infine per il processo di sviluppo è stato adottato anche il metodo del Weekly Cycle.

Questo metodo consiste nel proporre feature ed implementazioni alla committenza per soddisfare le sue richieste in maniera iterativa: regolarmente¹ si intrattiene una discussione coi committenti per stabilire quali aggiunte e sviluppi effettuare, e in seguito si fornisce alla committenza una nuova versione del progetto aggiornata e testata sugli aspetti richiesti.

Il fornire release funzionanti regolarmente ha consentito di andare incontro alla necessità dei filologi, così permettendo loro di presentare il progetto di PhiloEditor 7 anche durante la sua implementazione.

6.2 Participatory design

Come si può facilmente intuire dal precedente capitolo, il rapporto con la committenza durante lo sviluppo di PhiloEditor 7 è stato regolare e diretto. Questo perché, per fare in modo di venire incontro alle necessità dei filologi, si è optato per coinvolgere la committenza stessa nel progetto mediante riunioni e colloqui, seguendo il metodo del participatory design.

«Participatory Design (PD) represents a new approach towards systems design in which the people destined to use the system play a critical role in designing it.» [20]

«(Partecipatory design) rejects assumption that the goal of

¹Solitamente una volta alla settimana.

computerization is to automate the skills of human workers, instead seeing it as an attempt to give workers better tools for doing their jobs.» [20]

Le varie riunioni e i frequenti contatti e scambi coi filologi hanno permesso di sviluppare l'applicazione in una direzione certamente più chiara e mirata di quanto avrebbe potuto fare una semplice consegna iniziale all'avvio del progetto, senza ulteriori contatti.

Le prime riunioni con la committenza hanno visto come tema centrale la semplice ristrutturazione grafica del sito. Infatti, come esposto nel paragrafo 2.1.2, la componente UI/UX non consentiva una facile fruizione dell'applicazione da parte dei filologi. Questo ha portato alla creazione dei mockup iniziali, di cui i due esempi mostrati in Figura 6.1 e in Figura 6.2.

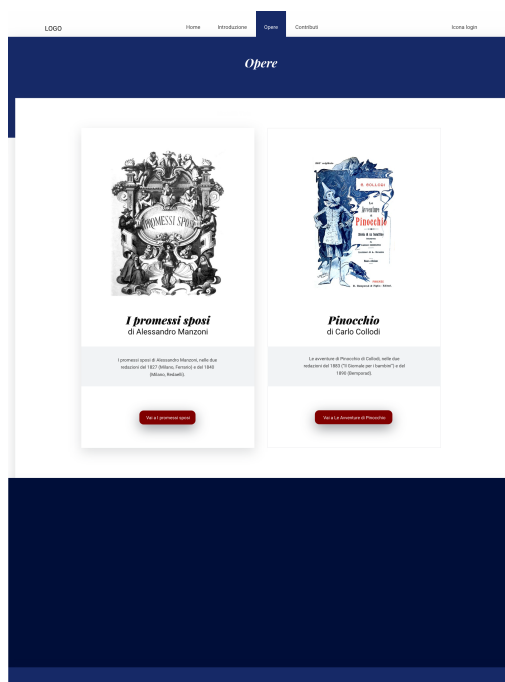


Figura 6.1: Mockup della sezione Opere.

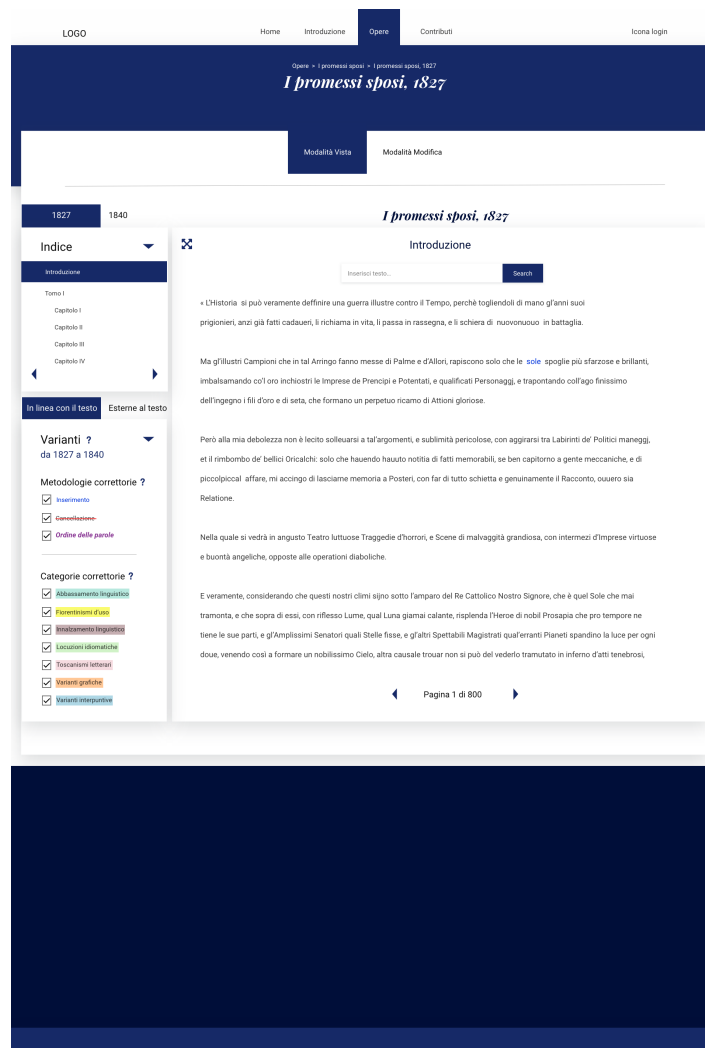


Figura 6.2: Mockup della sezione di lettura.

Successivamente, tali mockup sono stati mostrati alla committenza e, una volta approvati, sono stati realizzati. Come si potrà notare, nonostante l'approvazione iniziale, il design di PhiloEditor 7 è comunque cambiato in itinere. Questo grazie allo stretto rapporto

coi committenti: attraverso il dialogo tra team di sviluppo e filologi, è stato infatti possibile effettuare svariate modifiche, per assecondare le esigenze della committenza ed implementare al meglio le sue richieste.

Svolte le prime richieste, le riunioni hanno poi vertito sulle funzionalità del sito. Oltre alla necessità di risolvere numerosi bug nelle funzionalità preesistenti, sono state avanzate richieste e proposte su nuove funzionalità, quali la nuova modalità lettura e la modalità di ricerca sulle opere.

Anche in questo caso, il dialogo e il continuo fornire release funzionanti con piccole porzioni di codice aggiuntivo, hanno permesso uno sviluppo mirato, che assecdasse al meglio le esigenze del cliente. La committenza infatti s'è fatta carico, anche in questa fase dello sviluppo, di fornire un continuo feedback, che ha permesso di direzionare al meglio gli sforzi del team.

Capitolo 7

Valutazioni

7.1 Efficienza

PhiloEditor 7 a livello di efficienza si è dimostrato certamente una versione superiore rispetto alle precedenti, in particolare sui tempi di attesa da parte dell'utente.

Ora il sito carica la pagina iniziale più rapidamente rispetto alla precedente versione 6.5. Soprattutto da notare che non si verificano più casi di tempi di attesa infiniti o estremamente lunghi, dovuti ad alcuni bug all'interno delle funzioni di dimensionamento delle varie porzioni del sito.

Inoltre sono stati migliorati i tempi di caricamento della funzione di diffing, in particolare per quanto riguarda il confronto applicato su I promessi sposi. Infatti la suddivisione per capitoli e non per tomi ha alleggerito notevolmente il carico di lavoro della funzione sopracitata. Tutto ciò ha reso l'applicazione più snella e dinamica agli occhi dell'utente.

Degne di nota anche le tempistiche di elaborazione dei risultati della ricerca sulle opere. Infatti, il pacchetto TNTSearch, opportunamente modificato (paragrafo 4.3.2), ha dato risultati ottimi in fatto di velocità, restituendo anche parole con un alto numero di occorrenze in brevissimo tempo (Tabella 7.1). Un buon risultato, considerando che la ricerca viene effettuata su tutto il testo di tutte le edizioni di tutti i capitoli di tutte le opere all'interno di PhiloEditor.

Tabella 7.1: Velocità della ricerca sulle opere, risultati in millisecondi

Ricerca	Renzo (1147 occorrenze)	Lucia (787 occorrenze)	Pinocchio (775 occorrenze)	Geppetto (136 occorrenze)
1	292	57	139	88
2	277	56	134	76
3	234	56	134	76
4	218	53	143	93
5	250	47	140	74
6	220	46	135	73
7	238	48	152	70
8	225	47	130	77
9	271	51	176	75
10	281	51	135	82
Media	250.6	51.2	141.3	78.2

7.2 Efficacia

La nuova versione 7 di PhiloEditor ha aumentato anche l'efficacia della piattaforma nel fornire un valido supporto digitale ai filologi nel loro studio.

In particolare la nuova versione presenta un netto miglioramento nella fruibilità del sito, totalmente ridisegnato per permettere una facile navigazione anche ad un utente che abbia poca familiarità sia con la filologia che con la navigazione web.

PhiloEditor 7 è stato realizzato per fornire un'esperienza semplice, elegante e dinamica. La suddivisione in sezioni fa in modo che l'utente possa pian piano introdursi nell'esperienza, in maniera chiara e guidata. Stessa cosa vale per le modalità di lettura e marcatura, ripulite e attrezzate di tutto e non più del necessario per una buona fruizione dei contenuti. Inoltre la nuova sezione di tutorial permette, grazie a video dimostrativi

e spiegazioni scritte, di fugare la maggior parte dei dubbi che un nuovo utente potrebbe avere.

Sempre in tema di neofiti, anche la nuova modalità lettura è sicuramente un'ottima aggiunta che migliora la nuova edizione. Infatti permette anche a studenti non universitari di avvicinarsi alla filologia, grazie alla possibilità di leggere una singola edizione potendola mettere a confronto con l'altra semplicemente selezionando la modalità di confronto. Tutto questo fornisce la possibilità di un avvicinamento graduale allo studio filologico anche ad un utente inesperto.

La possibilità di effettuare la ricerca di parole chiave nelle opere e poterle poi visualizzare le occorrenze, fornisce invece un valido strumento al filologo esperto. Potrà infatti effettuare ricerche rapide e visualizzarne facilmente i risultati, permettendo confronti ad un livello superiore rispetto al semplice confronto fra capitoli.

La semplicità che PhiloEditor 7 crea anche nell'utilizzo dei vari tool quali le statistiche o la possibilità di marcare e salvare i testi, oltre che condividerli, visualizzarli ed esportarli, completano l'applicazione dandole tutto il necessario per essere un valido ed efficace strumento.

Questa pagina è lasciata intenzionalmente bianca.

Capitolo 8

Conclusioni e sviluppi futuri

La nuova versione di PhiloEditor ha seguito uno sviluppo basato sul dialogo, tra membri del team e la committenza. Questo ha portato alla creazione di PhiloEditor 7, uno strumento digitale dedicato a tutti i filologi e a coloro che desiderano approcciarsi allo studio della filologia, basato sulle esigenze espresse dai filologi stessi.

La versione 7 si presenta completamente funzionante e pronta all'uso, tanto che è stato possibile mostrarla sia al pubblico di filologi dell'Alma Mater Studiorum di Bologna che a quello de La Sapienza di Roma, riscontrando successo ed apprezzamenti in entrambi i casi. A riprova di quanto suddetto la versione 7 è stata inserita nella lista dei progetti attivi del /DH.ARC (Digital Humanities Advanced Research Centre), sotto il nome di PhiloEditor [21].

L'interfaccia semplice e fruibile e l'insieme di funzionalità, aggiunte, migliorate o corrette, basate sui feedback della committenza, compongono un'applicazione mirata, evoluta e valida nell'ambito della moderna filologia, adatta sia ad utenti alle prime armi che a ricercatori esperti.

PhiloEditor 7, nonostante sia completo e funzionante, può essere visto come una base per un'ambiente di studio filologico ancora più ampio.

Alcune future aggiunte dunque potranno certamente espandere e migliorare il sistema.

Un'aggiunta sicuramente utile potrà essere quella di permettere all'utente di creare le proprie categorie correttive, con relativo colore, per la marcatura dei testi. Quando accede alla modalità laboratorio dunque l'utente avrà a disposizione alcune categorie "standard" alle quali poter aggiungere le proprie, le quali dovranno essere salvate all'interno di un file dedicato che memorizzi tali personalizzazioni.

Poiché la lettura a schermo per tempi prolungati può affaticare la vista, potrebbe essere una valida aggiunta quella dell'opzione per un tema cromatico alternativo, che permetta di leggere i testi su sfondo scuro con scritte grigio-bianche, così da alleggerire il carico di stress visivo che subisce l'utente.

Siccome la funzione di ricerca sulle opere si basa su un file `.index` generato a priori, nel caso si vada nella direzione di un'aggiunta seriale o frequente di opere, potrebbe certamente facilitare gli inserimenti l'automatizzazione della creazione di tale file.

Per favorire inoltre la fruizione del sito a scopo didattico da parte di studenti non universitari, sarebbe interessante lo sviluppo di una nuova sezione dedicata agli esercizi (marcatura, quiz, risposte aperte, ecc.). Una volta completata l'esercitazione, l'insegnante potrà sfruttare la funzione di condivisione dei file, opportunamente modificata, per poter visualizzare gli elaborati dei suoi studenti.

Tale sviluppo richiederebbe quasi certamente la possibilità di avere diversi livelli di utenza, da quello "studente", limitato alla semplice lettura e marcatura a quello "amministratore", con la facoltà di aggiungere nuove opere, il tutto tramite un processo automatizzato che andrebbe costruito a sua volta.

Un'ultima aggiunta possibile potrebbe essere quella di migliorare ulteriormente la user experience, creando un tutorial interattivo che permetta al

nuovo utente di imparare come utilizzare la piattaforma, seguendolo passo passo all'interno delle funzionalità di cui PhiloEditor dispone.

Si concludono così le ipotesi attuali per i futuri sviluppi legati a PhiloEditor, suscettibili naturalmente ad un possibile incremento, in base alle nuove esigenze degli studiosi, ed altri utenti in generale, che emergeranno con l'uso dell'applicazione migliorata di PhiloEditor 7.

Questa pagina è lasciata intenzionalmente bianca.

Bibliografia

- [1] Giovanni Treccani et al. *Enciclopedia Italiana di scienze, lettere ed arti*. Vol. 15. Istituto dell'Enciclopedia Italiana fondata da G. Treccani, 1994.
- [2] Claudia Bonsi, Angelo DiIorio, Paola Italia, Francesca Tomasi, Fabio Vitali e Ersilia Russo. «PhiloEditor®: simplified HTML markup for interpretative pathways over literary collections». In: Università Cattolica del Sacro Cuore, 2017, p. 1. URL: https://convegni.unicatt.it/aiucd-Bonsi_et_al.pdf.
- [3] Paola Italia e Claudia Bonsi. *Edizioni Critiche Digitali Digital Critical Editions: Edizioni a confronto Comparing Editions*. Vol. 34. Sapienza Università Editrice, 2016.
- [4] Friedrich Wilhelm Nietzsche. *Aurora. Pensieri sui pregiudizi morali*. Vol. 328. Newton Compton Editori, 2012.
- [5] Autori vari. *Filologia*. URL: <http://en.wikipedia.org/wiki/Filologia>. (accessed: 2.07.2020).
- [6] Autori vari. *Filologia Classica*. URL: http://en.wikipedia.org/wiki/Filologia_classica. (accessed: 2.07.2020).
- [7] Giovanni Treccani. *Vocabolario della lingua italiana*. Istituto dell'Enciclopedia Italiana fondata da G. Treccani, 1991.
- [8] Luca Carlo Rossi. *Finalità e metodi della filologia*. ICoN – Italian Culture on the Net, 2003, pp. 22–26.

- [9] Patrick Sahle, Georg Vogeler e IDE (Institut für Dokumentologie und Editorik). *Criteria per la recensione delle edizioni digitali scientifiche (EDS), versione 1.1*. 2014. URL: www.i-d-e.de/publikationen/weitereschriften/criteri-versione-1-1/.
- [10] Cacycle. *User:Cacycle/diff*. URL: <http://en.wikipedia.org/wiki/User:Cacycle/diff>. (accessed: 27.06.2020).
- [11] TEI Consortium. *TEI: Text Encoding Initiative*. URL: <http://tei-c.org>. (accessed: 27.06.2020).
- [12] Autori vari. *Text Encoding Initiative*. URL: http://it.wikipedia.org/wiki/Text_Encoding_Initiative. (accessed: 27.06.2020).
- [13] GO-Globe. *The Importance Of Professional Web Design For Your Business Website*. 15.05.2020. URL: www.go-globe.hk/importance-of-professional-web-design/.
- [14] TNT Studio. *TNTSearch*. URL: <http://github.com/teamtnt/tntsearch>. (accessed: 27.06.2020).
- [15] TNT Studio. *Solving the search problem with Laravel and TNTSearch*. URL: <http://tnt.studio/blog/solving-the-search-problem-with-laravel-and-tntsearch>. (accessed: 27.06.2020).
- [16] Agile Alliance. *Extreme Programming Manifesto: Definition*. URL: www.agilealliance.org/glossary/xp. (accessed: 27.06.2020).
- [17] Agile Alliance. *Extreme Programming Manifesto: Values*. URL: www.agilealliance.org/glossary/xp. (accessed: 27.06.2020).
- [18] Kent Beck. *Extreme programming explained: embrace change*. Addison-Wesley Professional, 2000, p. 20.
- [19] Agile Alliance. *Extreme Programming Manifesto: Practices*. URL: www.agilealliance.org/glossary/xp. (accessed: 27.06.2020).
- [20] Douglas Schuler e Aki Namioka. *Participatory design: Principles and practices*. CRC Press, 1993.
- [21] /DH.ARC. *Research: PhiloEditor*. URL: <http://centri.unibo.it/dharc/en/research/projects-at-dh-arc>. (accessed: 27.06.2020).

Elenco delle figure

3.1	Visualizzazione tradizionale di un sistema di diffing.	15
3.2	Visualizzazione sovrapposta applicata da PhiloEditor.	15
3.3	Esempio di marcatura del testo.	16
3.4	Confronto fra gli strumenti in modalità Vista e modalità Modifica	18
3.5	Ricerca nel testo.	19
3.6	Hide del bottone Modifica in caso di mancato login.	21
4.1	Sezione di benvenuto (home).	26
4.2	Sezione introduttiva (home).	26
4.3	Sezione delle opere disponibili (home).	27
4.4	Sezione di introduzione.	28
4.5	Sezione delle opere.	29
4.6	Sezione di un'opera specifica.	30
4.7	Modalità Varianti.	31
4.8	Modalità Lettura.	32
4.9	Modalità Laboratorio.	33
4.10	Visualizza.	35
4.11	Modifica.	36
4.12	Sezione dei risultati di ricerca sulle opere.	37
4.13	Visualizzazione dell'occorrenza selezionata.	37
6.1	Mockup della sezione Opere.	56
6.2	Mockup della sezione di lettura.	57

Elenco delle tabelle

7.1	Velocità della ricerca sulle opere, risultati in millisecondi . . .	60
-----	---	----

Listings

5.1	Esempio di un output di ricerca.	45
5.2	Struttura della cartella sources.	46
5.3	Struttura della cartella contributors.	47
5.4	Esempio di un risultato per la ricerca effettuata sulla parola “renzo” con le rispettive occorrenze.	49