

# Progetto Codifica di testi

William Brai - Jacopo Gentili

## Introduzione

Il progetto è stato realizzato nella versione TEI P5, usando la dtd appositamente studiata per i manoscritti ovvero `tei_ms.dtd`.

Abbiamo selezionato e codificato le pagine 7 e 8 (immagini 17 e 20) delle “*Trois premières conférences à l'Université*”.

## Codifica XML

La root ovviamente sarà il tag TEI che avrà come figli: il `teiHeader`, `text`, `facsimile`.

`TeiHeader` avrà a sua volta dei figli:

- ❖ **fileDesc**: contiene le informazioni sul documento digitale che si è creato, i suoi figli sono:
  - **titleStmt**: informazioni riguardanti il titolo del file e i suoi responsabili (codifica e trascrizione).
  - **editionStmt**: informazioni relative alla nostra edizione digitale
  - **publicationStmt**: Informazioni relative alla pubblicazione del file.
  - **sourceDesc**: Descrizione del manoscritto originale e delle fonti bibliografiche utilizzate per la codifica. l'elemento **bibl** è stato utilizzato per racchiudere le informazioni del riferimento bibliografico, mentre **MsDesc** per una dettagliata strutturazione degli elementi appartenenti al manoscritto originale (titolo, composizione, provenienza, dimensione e informazioni su come è stato scritto) .
- ❖ **encodingDesc**: informazioni relative alla codifica
- ❖ **profileDesc**: Sono state inserite le lingue utilizzate nella codifica

L'elemento principale `text` ha come figlio un **group** che a sua volta avrà due elementi `text`, il primo indicherà la pagina 7 del manoscritto e il secondo la pagina 8.

Per quando riguarda la pagina 7, dopo il **body**, si aprono 3 **div**:

1. Il primo conterrà il testo del manoscritto
2. il secondo ha al suo interno una lista di glosse trovate nel dizionario `Simple_fds` (quelle in francese) e sul web
3. il terzo contiene le note trovate sul PDF “*Lingua e mente sociale*”

Stesso discorso per la pagina 8.

Dopo le note è presente una lista per far riferimento ai nomi (**ListPerson**).

Il testo del manoscritto (soprattutto nella pagina 7) presenta un gran numero di cancellazioni ed aggiunte dello stesso De Saussure che abbiamo codificato con i tag **add**, **del**, **gap**,

**unclear** con i loro rispettivi attributi, in base alla necessità. **Gap** è stato utilizzato soprattutto perché non riuscivamo a leggere le parole cancellate.

Nel testo è stato usato il tag **<lb/>** per indicare l'inizio di una nuova linea, racchiusa dentro ad un elemento **<seg>** che indica il contenuto della linea.

A seguire possiamo trovare un div contenente le glosse, realizzate con **<gloss>** e **<term>** le quali faranno riferimento ai **<term>** presenti nel testo, collegate tramite l'attributo **ref**.

Le glosse sono contenute in una lista che a sua volta contiene degli **Item**. A ciascun item corrispondono delle coppie **<term>**, **<gloss>**.

Per le note (**<note>**) è stato creato un puntatore **<ptr>** che rimanda appunto alle rispettive note tramite l'attributo **target**.

Infine è presente l'elemento **<facsimile>**, al suo interno troviamo 2 **<surface>** per entrambe le pagine e a loro volta contengono l'immagine di riferimento e degli elementi **<zone>** con un **xml:id** e le coordinate delle varie linee ottenute grazie a TEI Zoner. Le zone sono state collegate con gli elementi **<lb>** tramite l'attributo **facts**.

## Trasformazione XSL

### visualizzazione e navigazione

Nel foglio di stile xsl è stato inserito codice html per costruire una pagina utilizzando gli apply-template. Volta volta abbiamo scelto gli elementi che volevamo far visualizzare nella pagina html e ne abbiamo fatto un match. Di conseguenza avremo:

- Titolo + autore (inseriti all'inizio del body)
- Descrizione del manoscritto (passando in output valori contenuti in **MsDesc**)
- Collegamenti bibliografici dell'edizione francese di riferimento e del libro da cui si sono prese le note. (ottenuti passando gli elementi contenuti in **listBibl**).
- Immagine e testo di pagina 7 e 8 con l'aggiunta di una funzione JavaScript che permette di visualizzare la traduzione italiana. (prendendo il valore di **graphic** contenuto in **surface**).
  - Abbiamo creato delle mappe contenenti delle aree con coordinate e id equivalenti a quelle delle zone del nostro documento xml. Il procedimento è stato svolto creando delle variabili xml e fornendo agli attributi delle aree i valori corrispondenti delle variabili xml. A specchio abbiamo creato dei collegamenti **<a>** (con attributo **id**) nel testo per identificare la posizione degli **<lb/>** presenti nel documento xml (gli **<a>** sono stati creati grazie al template **dividi** che oltre a creare dei collegamenti nelle posizioni degli **<lb>** fa anche andare a capo il testo con dei **<br />**). Infine abbiamo collegato, in maniera reciproca, le aree con gli **<a>** mediante l'attributo **href**. es(**<area id=l1\_p7 href="##l1">** **<a id=#l1\_p7 href="#l1\_p7">**). Malgrado il codice non siamo riusciti a scrivere la funzione javascript per visualizzare in output la corrispondenza tra le aree dell'immagine e gli **<a>** del testo relativi alle linee. Di conseguenza gli **Lb** sono stati nascosti attraverso il css.

- Le inserzioni di note e glosse sono state effettuate attraverso template che forniscono il valore degli elementi presenti nel documento xml: **gloss**, **term** e **note**.
  - Grazie ai template: **ancora7** e **ancora8** abbiamo realizzato ed inserito degli elementi `<a>` riferiti ai collegamenti `<ptr>` dei vari `<term>` presenti nel testo del documento xml.
  - I template **ancornote7** e **ancornote8** invece hanno realizzato dei collegamenti `<a>` nei div che contengono le note.
  - Riassumendo abbiamo collegato reciprocamente nel testo gli `<a>` (ottenuti tramite il template **ancora**) con gli `<a>` contenuti nei div relativi alle note grazie all'attributo **href** (ottenuti tramite il template **ancornote**).
- Una footer con il link dell'archivio digitale, la sua relativa licenza e i responsabili dell'edizione francese, corrispondente al testo francese presente nel PDF prolusioni

## Funzionamento

Il file XML è stato validato con il seguente comando:

```
java -cp "./Xerces-J-bin.2.12.1/xerces-2_12_1/*" dom.Counter -v ProgettoSauss.xml
```

E la generazione della pagine html con questo:

```
java -jar SaxonHE10-3J/saxon-he-10.3.jar -s:ProgettoSauss.xml -xsl:ProgettoSauss.xsl -o:testSauss.html
```

```
jacop@LAPTOP-RUICMEBM MINGW64 ~/OneDrive/Desktop/codifica di testi/corsoCodifica-master/CTaa20-21/src
$ java -cp "./Xerces-J-bin.2.12.1/xerces-2_12_1/*" dom.Counter -v ProgettoSauss.xml
ProgettoSauss.xml: 347;50;1 ms (505 elems, 1542 attrs, 5869 spaces, 18667 chars)

jacop@LAPTOP-RUICMEBM MINGW64 ~/OneDrive/Desktop/codifica di testi/corsoCodifica-master/CTaa20-21/src
$ java -jar SaxonHE10-3J/saxon-he-10.3.jar -s:ProgettoSauss.xml -xsl:ProgettoSauss.xsl -o:testSauss.html
```