# Programma Tesi

*(note sul programma della tesi)*

#Librerie

Nel programma vengono usate le seguenti librerie:
1. BeautifulSoup o bs4.
2. Requests.
3. Openpyxl.
4. Datetime.
5. Os.
6. System.
7. Msvcrt.
8. Random (solo per il test).

#Funzioni

Nel programma sono presenti le seguenti funzioni:
1. Funzione test.
2. Funzione initialize.
3. Funzione searching.
4. Funzione scraping.
5. Funzione update.
6. Funzione save.
7. Funzione logerror.
8. Funzione table_write.
9. Funzione remove_link.

#Struttura

La struttura del programma può esser riassunta in:
1. Controllo sessioni precedenti.
2. Ricerca.
3. Estrapolazione dati.
4. Apertura link con errore.
5. Aggiornamento Database.

# Funzione test

#Definizione

La funzione 'test' eliminerà a caso delle pagine contenute nella variabile 'lett' (vedi Funzione initialize) finché la sua dimensione non sarà uguale al valore indicato da 'num' (attualmente 15 ma modificabile).

#Param

Parametri:
-*num*: valore ridimensionamento di lett.
- *lett*: lista che contiene l'elenco delle pagine trovate dalla funzione initialize.
Return:
-*lett*: lista che contiene l'elenco delle pagine trovate dalla funzione initialize con resize a num.

# Funzione initialize

#Definizione

La funzione 'initialize' serve per recuprare l'elenco di tutte le pagine del vocabolario. Il vocabolario è consultabile a blocchi al link https://www.um.es/lexico-comercio-medieval/index.php/mapaweb sotto la voce "Vocabulario". I vocaboli sono divisi in blocchi per lettera iniziale. Ogni blocco contiene una serie di pagine che contengono tutte le voci che iniziano per una determinata lettera.

#Param

Parametri:
- *None*.
Return:
- *res*: lista che contiene l'elenco delle pagine trovate.

# Funzione searching

#Definizione

La funzione 'searching' serve per estrapolare i vocaboli presenti nel sito https://www.um.es/lexico-comercio-medieval/. La funzione analizza i risultati proposti dal sito ed estrae i parametri '*href*' tramite:
- *risultati = soup.find('article', class_='resultados')*: viene selezionata l'area di ricerca.
- *a_risultati = risultati.find_all('a')*: vengono salvati tutti i tag 'a' (HTML link) in una lista.
- *vrf = str(i.get('href'))*: si estrae il valore di 'href' (link/indirizzo vero e proprio).

Questi poi vengono salvati in una lista e restituiti al chiamante.

#Param

Parametri:
- *pref:* variabile contenente il prefisso dell'url delle pagine di ricerca.
- *s_link:* contiene l'url della sezione di ricerca del sito senza la query string completa.

Return:
- *res*: lista contenente tutti i link trovati dal campo di ricerca.

# Funzione scraping

#Definizione

La funzione 'scraping' estrae le informazioni rilevanti per ogni vocabolo. Vengono estratti:
1. *Vocabolo*: la parola interessata.
2. *Definizione*: la descrizione del termine.
3. *Tipo*: la tipologia del termine.
4. *Documenti Referenziati*: eventuali reference legate alla parola.
5. *Archivio Documenti Referenziati*: contiene tutti i documenti che sono stati referenziati.
6. *Numero di Documenti Referenziati*: la somma di tutti i documenti referenziati.
7. *URL:* link della pagina.
8. *Data di Salvataggio:* quando il record è stato salvato

I dati vengono raccolti in una lista e poi restituiti.

#Param

Parametri:
- *l*: link della pagina da analizzare.

Return:
- *res*: lista contenente i dati estratti.

#Nota

La funzione esegue una serie di controlli sulla risposta del server all'interrogazione da parte del programma.
- Blocco reindirizzamenti a siti esterni: l'accesso ad un sito esterno renderebbe vana l'estrapolazione dei dati in quanto il programma è stato tarato per lavorare solo su un determinato sito.
- Verifica dello *status code* a seguito della richiesta: se lo status code restituito da '*response.status_code*' è diverso da 200 (*200 OK*) vuol dire che la pagina richiesta non è pronta per la lettura. In tal caso viene lanciata l'ecezione **BeautifulSoupException → possibile aggiunta a Funzione logerror in caso di status code diverso *(per completezza)***.

# Funzione update

#Definizione

La funzione 'update' serve per eseguire l'aggiornamento (se presente) delle voci che sono state salvate nel foglio excel. Questo controllo viene eseguito durante la fase di Aggiornamento Database. La funzione riapre il collegamento alla pagina indicata dal parametro 'link' e controlla se sia presente all'interno della pagina la data che indica l'ultima modifica effettuata. Se presente, la funzione controlla che la data di salvataggio della voce nel database sia antecedete la data riportata nel sito e se sì procede ad una nuova estrazione dei dati richiamando Funzione scraping. Infine, chiamando la Funzione table_write sovrascrive i dati nel foglio excel nella stessa identica posizione dell'originale.

#Param

Parametri:
-*rdate*: data di salvataggio del record nel database.
-*link*: link della pagina.
-*worksheet*: file excel.
-*row*: riga dove il vocabolo è stato salvato.
Return:
-*None*.

# Funzione save

#Definizione

La funzione save serve per salvare all'interno del file .txt l'ultimo record salvato o l'ultima ricerca effetuata in modo che, in caso di arresto improvviso del programma, sia possibile recuperare la precedente sessione. → vedi Controllo sessioni precedenti per maggiori spiegazioni
La funzione apre il file indicato dal parametro *s_path*, ne cancella il contenuto e successivamente scrive il valore contenuto nel parametro *link*.

#Param

Parametri:
- *s_path*: il relative path del file dove effettuare il salvataggio.
- *link*: il link da salvare → **si potrebbe cambiare il nome in param**.

Return:
- *None*.

#Nota

Usando '*file.seek(0)*' viene portato il puntatore al file in posizione 0 (all'inizio) e con '*file.truncate(0)*' viene fatto un resize del file a 0 Kbyte che di fatto cancella il contenuto del file. Per entrambe le funzioni serve accedere al file in modalità '*r+*' ovvero in modalità di lettura/scrittura con puntatore all'inizio del file.

# Funzione logerror

#Definizione

La funzione 'logerror' serve per salvare nel file 'logerrori.txt' tutti gli errori che sono avvenuti durante il funzionamento del programma. Vengono principalmente scritti errori derivati dal lancio di eccezioni durante il programma. I dati vengono scritti come *data + eccezione*.

#Param

Parametri:
- *l*: variabile che contiene il link della pagina interessata da errore.
- *exception* : contiene l'eccezione lanciata.

Return:
- *None*.

# Funzione table_write

#Definizione

La funzione 'table_write' serve per scrivere i dati estrapolati dalla funzione Funzione scraping su un file excel. I risultati una riga alla volta e corrispondono a tutte le informazioni estratte per un solo vocabolo.

#Param

Parametri:
- *res*: contiene i dati estratti dalla funzione scraping (vocabolo, definizione, tipo, documenti. referenziati, archivio documenti, numero documenti referenziati ed url).
- *worksheet*: contiene la pagina di lavoro excel sulla quale verranno scritti i record.
- *r*: contatore della riga di scrittura.

Return:
- *None*.

# Funzione remove_link

#Definizione

La funzione 'remove_link' viene utilizzata durante la fase di Estrapolazione dati. La funzione rimuove dal file 'links.txt' un link che ha avuto un problema di connessione. Questo viene fatto perché in caso di arresto anomalo del programma il dato contenuto in 'lastrecord.txt' sia consistente con 'links.txt' (perché se non venisse rimosso e si arrestasse ad esempio all'iterazione successiva, il file 'links.txt' segnalerebbe come salvato il link affetto da errore di connesione → **causerebbe una inconsistenza dei dati**).
Al termine della rimozione viene chiamata la Funzione logerror se viene passata una eccezione.

#Param

Parametri:
- *link*: contiene il link da rimuovere.
- *exception*: contiene l'eccezione.

Return:
- *None*.

# Controllo sessioni precedenti

#Definizione

In questa fase il programma verifica il contenuto dei file 'lastlink.txt' e 'lastrecord.txt' per poter ripristinare la sessione precedente.
1. Inizialmente il programma controlla il contenuto di 'lastrecord.txt'. Se il file contiene un link e l'utente desiderasse continuare con la sessione precedente, il programma procederà alla fase di Estrapolazione dati e continuerà con il salvataggio dei record a partire da quello indicato dal file.
Il programma apre il file 'links.txt', cerca il link contenuto in 'lastrecord.txt' e lo salva nella variabile lastrecord. Successivamente copierà nella variabile link_risultati i valori di 'links.txt' dal indice di lastrecord fino alla fine.
*index = old_link.index(lastrecord)*
*for i in range(index+1, len(old_link)):*
*link_risultati.append(i)*
In caso contrario eliminerà il contenuto di 'lastrecord.txt' ed eliminerà tutti i link contenuti in 'links.txt' che non sono stati salvati → **questo per mantenere il database consistente**
2. Se 'lastrecord.txt' è vuoto allora il programma procederà a controllare il contenuto di 'lastlink.txt'. Se contiene un link e l'utente desiderasse contiuare con la sessione precedente, allora il programma continuerà la fase di Ricerca a partire da dove era stata interrotta. Verrà quindi ricalcolato il punto di partenza della ricerca.
Il programma salva in 'lastlink' il contenuto del file 'lastlink.txt' e procede come la modalità classica. Dopo la chiamata della Funzione initialize, il programma rimuoverà le pagine già analizzate (ovverto tutte quelle precedenti a 'lastlink')
3. Se entrambi dovessero essere vuoti il programma partirà in modalità classica seguendo tutte le fasi descritte nel Programma Tesi.

#Pseudocodice

*if → il file 'lastrecord.txt' non è vuoto:*
*true → chiedo se utente vuole continuare*
*if → utente vuole continuare:*
*true → sezione salvataggio excel, elimino contenuto 'lastrecord.txt'*
*false → elimino 'lastrecord.txt'*
*false → controllo 'lastlink.txt'*
*if → il file 'lastlink.txt'non è vuoto:*
*true → chiedo utente se vuole continuare*
*if → utente vuole continuare:*
*true → ricalcolo gli indici 'in_a1' e 'in_a2'*
*false → modalità classica*
*false → modalità classica*

# Ricerca

#Definizione

La fase di ricerca sfrutta l'Algoritmo di ricerca per trovare tutti i vocaboli contenuti nel sito https://www.um.es/lexico-comercio-medieval/index.php/p/v/inicio. E' costituita dalle seguenti sotto-fasi:

1. Viene invocata la Funzione initialize per recuperare le pagine che contengono tutti i vocaboli. Il tutto viene salvato nella variabile 'lett'.
2. Per ogni elemento contenuto in 'lett' viene invocata la Funzione searching per recuperare tutti i vocaboli contenuti in una determinata pagina. Il tutto viene salvato nella variabile 'res'.
3. Per ogni elemento contenuto in 'res' viene controllato che non sia già stato inserito nel database (controllando la variabile 'old_link') e che non sia già stato inserito nella variabile 'link_risultati' (che contiene tutti i link a vocaboli che successivamente verranno elaborati)
4. Viene salvato in 'lastlink.txt' il link dell'ultima ricerca fatta

In caso di errore di connessione il programma termina in quanto non è possibile continuare (il server non risponde correttamente).

# Algoritmo di ricerca

#Definizione

Tutti i vocaboli sono reperibili all'indirizzo: https://www.um.es/lexico-comercio-medieval/index.php/mapaweb.

La mappa del sito contiene un elenco (di link) che riportano a tutte le parole referenziate nel sito.

# Estrapolazione dati

#Definizione

La fase di estrapolazione dati viene eseguita in maniera ciclica per tutti i link trovati durante la fase di Ricerca. In questa fase, per ogni vocabolo, vengono recuperati i seguenti dati:

1. Vocabolo: la parola in se.
*v = soup.find('h3', class_='lexema_title')*
2. Definizione: ogni vocabolo è accompagnato da una descrizione che ne spiega il significato. Per rendere il testo estratto omogeneo, vengono ignorati tutti gli a capo.
*df = soup.find('p',  class_='descripcion')*
3. Tipo: ogni vocabolo è catalogato con un tipo.
*t = soup.find('span', class_='tipo')*
4. Documenti Referenziati 1: i vcaboli possono essere collegati ad alcuni scritti del professor Gual. Qui viene indicato se il vocabolo è stato referenziato in qualche scritto e come.
*rf = soup.find('div',  id='detalle_imagenes_lexema')*
5. Archivio Documenti Referenziati 1: l'elenco dei documenti che sono stati referenziati dal vocabolo.
*arch = soup.find('div', id='imagenes')*
6. Documenti Referenziati 2: i vocaboli che sono collegati agli scritti preparati dal gruppo di ricerca del Juan March Foundation Grant (il cui autore appare in fondo).
*rf = soup.find('div', id=None)*
7. Archivio Documenti Referenziati 1: l'elenco dei documenti che sono stati referenziati dal vocabolo.
*arch = soup.find('div', id='imagenes_jmarch')*
8. Numero Documenti Referenziati: il numero di scritti a cui il vocabolo è collegato.
9. URL: il link della pagina che tratta il vocabolo.
10. Data di Salvataggio: il giorno in cui il record viene salvato.

#Nota

Se un link presenta un errore di connessione *(es TimeOutException)* allora il link viene rimosso da *links.txt* tramite la funzione Funzione remove_link e salvato nella variabile *link_error* per essere trattati successivamente (vedi fase Apertura link con errore).

# Apertura link con errore

#Definizione

Viene effettuata la fase di Estrapolazione dati per i link che hanno avuto un problema di connessione. I link che hanno avuto un problema in quella fase vengono salvati nella variabile *link_error*.
Con un ciclo *while* vengono riaperti tutti i link nella variabile *link_error*. Per ogni link viene rifatta la fase di Estrapolazione dati e, se si dovesse ripresentare un errore di connessione, il link viene messo in coda e riprocessato più avanti. Se l'estrapolazione dei dati avviene con successo allora il link viene reinserito nel file *links.txt* per esser registrato come salvato. Se si effettuano troppi tentativi di connessione avviene l'uscita dal ciclo e l'eliminazione della variabile *link_error*.

#Nota

I link che sono contenuti in *link_error* non sono contenuti nel file *links.txt* a meno che non vengano salvati correttamente.

# Aggiornamento Database

#Definizione

Ricontrollo per le voci già salvate nel database se la loro corrispondente versione online ha subito qualche modifica che quindi dev'essere riportata anche nel database locale.
Inizialmente recupero tutti i link contenuti nel file excel.

#Nota

Non utilizzo il file *links.txt* poiché in caso di modifica dell'ordine dei vocaboli *(es. si vuole mantenere sempre l'ordine alfabetico)* questi potrebbero non essere nello stesso ordine di quelli contenuti in *links.txt*.

Poi per ogni link che ho recuperato invoco la Funzione update che controlla se la data di modifica riportata nel sito è più recente di quella riportata nel database. In caso affermativo aggiorno il vocabolo corrispondente nel database, altrimenti passo al successivo.

#Nota

L'aggiornamento viene saltato per le voci che hanno data di scrittura nel database uguale alla data corrente. Questo controllo viene fatto prima d'invocare la Funzione update. Questo serve perché l'aggiornamento non controlli le voci appena salvate durante la fase di Estrapolazione dati.