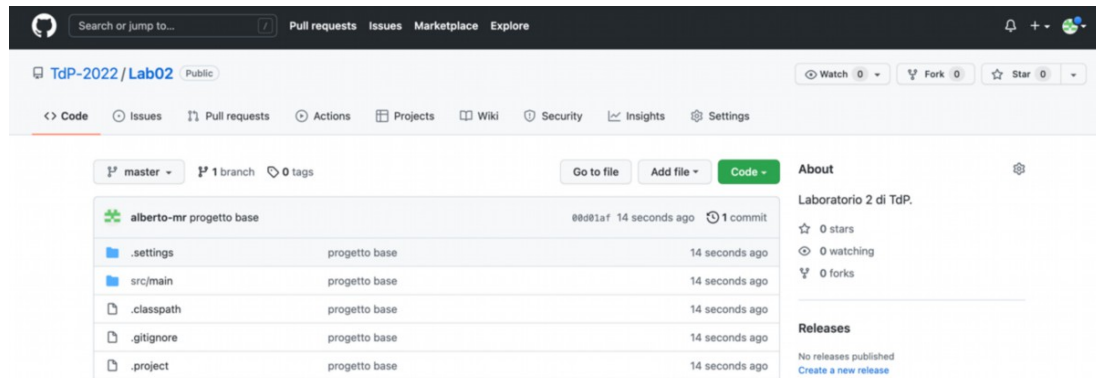


03FYZ TECNICHE DI PROGRAMMAZIONE

Istruzioni per effettuare il fork di un repository GitHub

- Effettuare il login su GitHub utilizzando il proprio username e password.
- Aprire il repository su GitHub relativo al quarto laboratorio:
<https://github.com/TdP-2025/Lab04>
- Utilizzare il pulsante *Fork* in alto a destra per creare una propria copia del progetto. L'azione di Fork crea un nuovo repository nel proprio account GitHub con una copia dei file necessari per l'esecuzione del laboratorio.



- Aprire Pycharm, assicurandosi che eventuali precedenti progetti siano chiusi, selezionare *Get From VCS*. Utilizzare la URL del **proprio** repository che si vuole clonare (**non** quello in TdP-2025!), ad esempio:

<https://github.com/my-github-username/Lab04>

- Selezionare la cartella di destinazione (quella proposta va bene), fare click su *Clone*. ■ Il nuovo progetto è stato clonato ed è possibile iniziare a lavorare.
- A fine lavoro ricordarsi di effettuare Git commit e push, utilizzando l'apposito menù.

ATTENZIONE: solo se si effettua Git **commit** e successivamente Git **push** le modifiche locali saranno propagate sui server GitHub e saranno quindi accessibili da altri PC e dagli utenti che ne hanno visibilità.

03FYZ TECNICHE DI PROGRAMMAZIONE

Esercitazione di Laboratorio 17/18 Marzo 2024

Obiettivi dell'esercitazione:

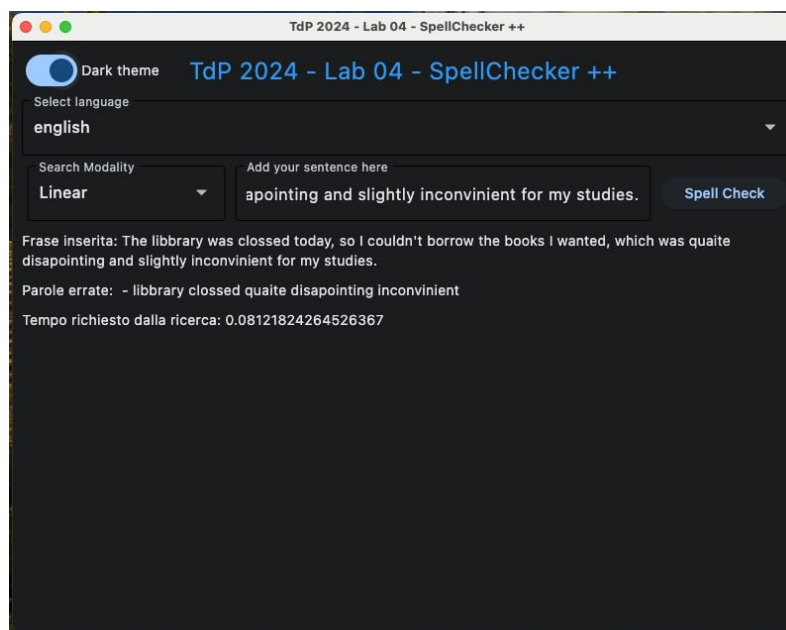
- Utilizzo pacchetto flet di Python
 - Utilizzo del Pattern MVC
-

ESERCIZIO 1

Dopo aver fatto il fork del progetto relativo al quarto laboratorio, realizzare in linguaggio Python un'applicazione dotata di interfaccia grafica che permetta la gestione del programma atto alla correzione delle parole sviluppato nello scorso laboratorio (Lab3).

L'applicazione permette di selezionare la lingua (Italiano/Inglese/Spagnolo), selezionare la modalità di ricerca (Contains/Linear/Dicotomica) e di inserire la frase da sottoporre all'algoritmo di correzione. L'applicazione dovrà poi restituire la frase inserita dall'utente, l'elenco delle parole errate e il tempo richiesto dalla ricerca.

Di seguito un esempio dell'interfaccia grafica da realizzare:



Fare uso del pattern MVC e del pacchetto **flet**, come spiegato a lezione.

Traccia per svolgere l'esercizio

- Il file controller.py contiene la classe SpellChecker, che può essere ereditata dal Lab03 con le opportune modifiche. Hint: il metodo handleSentence può essere modificato per accettare in ingresso un ulteriore campo "modality", in modo da selezionare dall'esterno l'algoritmo di ricerca richiesto.
- Il file view.py contiene una classe denominata View che si occupa esclusivamente di gestire l'interfaccia grafica.
- In tale classe definire i seguenti layout:
 - o Titolo pagina ("TdP 2025 – Lab 04 – SpellChecker++") – già fatto.
 - o Bottone per cambiare tema (default light mode) – già fatto.
 - o Row 1: inserire un menu a tendina (hint: pacchetto flet, control Dropdown) per selezionare la lingua con tre opzioni (italiano/inglese/spagnolo). Una volta selezionata l'opzione desiderata la funzione richiamerà un'altra funzione per verificare la corretta selezione della lingua (far comparire un messaggio nell'interfaccia se è avvenuta o meno la corretta selezione)
 - o Row 2 parte sinistra: creare un menu a tendina (hint: pacchetto flet, control Dropdown) per selezionare il tipo di ricerca con tre opzioni (Default/Lineare/Dicotomica). Una volta selezionata l'opzione desiderata verrà richiamerà un'altra funzione per verificare la corretta selezione (far comparire un messaggio nell'interfaccia se è avvenuta o meno)
 - o Row 2 parte centrale: creare uno spazio dove inserire il testo (hint: pacchetto flet, control TextField)
 - o Row 2 parte destra: creare un bottone per avviare la funzione di correzione ortografica (hint: pacchetto flet, control ElevatedButton). Questo bottone, una volta cliccato, richiamerà una funzione ("handleSpellCheck") in grado di avviare l'algoritmo di ricerca selezionato sulla frase inserita dall'utente. In questa funzione, prima di richiamare il metodo in grado di ritornare le parole sbagliate e il tempo impiegato a trovarle (funzione handleSentence() del lab03), è necessario inserire vari controlli su l'inserimento del linguaggio e della modalità di ricerca. Qualora mancasse uno o più campi, visualizzare un messaggio nell'interfaccia. Una volta eseguito correttamente l'algoritmo di ricerca, la funzione "handleSpellCheck" dovrà riportare sull'interfaccia: 1. La frase inserita, 2. Le parole errate e 3. Il tempo richiesto dalla ricerca. Infine, alla pressione del tasto, il metodo si dovrà occupare anche di svuotare il TextField che contiene la frase.
 - o Row 3: creare un'area di testo (hint: pacchetto flet, control ListView) in cui stampare gli output del metodo "handleSpellCheck" associato al tasto. ATTENZIONE alla sintassi di ListView. Per aggiungere del contenuto all'area di testo, è necessario fornire come ingresso un controller e non una stringa, ad esempio:

```
§lv = ft.ListView(expand=1, spacing=10, padding=20, auto_scroll=True)

count = 1

for i in range(0, 60):
    lv.controls.append(ft.Text(f"Line {count}"))    count += 1
```
- Le funzioni di handle che gestiscono l'interazione fra il modello e l'interfaccia grafica sono preferibilmente da istanziare come metodi del controller.