

03FYZPL 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/Lab05>
- 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/Lab05>
- 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.

03FYZPL TECNICHE DI PROGRAMMAZIONE

Esercitazione di Laboratorio 25/26 Marzo 2025

Obiettivi dell'esercitazione:

- Utilizzo del Pattern MVC
 - Utilizzo di mysql-connector-python
 - Introduzione al Pattern DAO
-

ESERCIZIO 1

Dopo aver fatto il fork del progetto relativo al quinto laboratorio, realizzare in linguaggio Python un'applicazione dotata di interfaccia grafica che permetta la gestione degli studenti e dei corsi di ateneo. L'applicazione permette di cercare informazioni relative ad un corso o ad uno studente e di iscrivere uno studente ad un corso. Prestare attenzione alla gestione di tutte le possibili condizioni di errore.

Di seguito si propone un esempio dell'interfaccia grafica

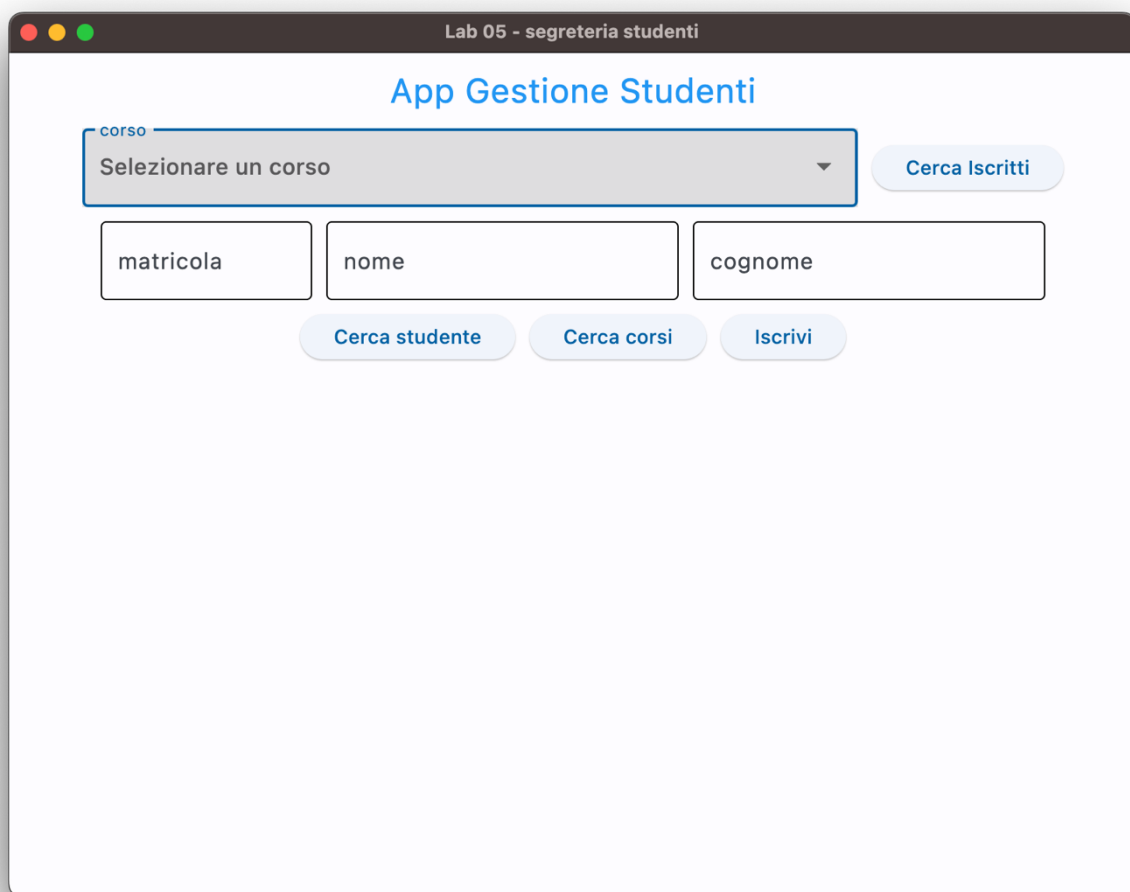


Figure 1: esempio di interfaccia grafica

Fare uso dei patterns **MVC** e **DAO**, usando i pacchetti **flet** e **mysql-connector-python**, come spiegato a lezione.

Traccia

Punto 1

Realizzare un'interfaccia grafica con *flet* simile a quella mostrata in *Fig 1*. La proposta di interfaccia include un menu **DropDown**, per **selezionare un corso**, ed un **TextField** per **specificare una matricola**. Sono previsti due ulteriori campi di testo non-editabili (`read_only=True`) per visualizzare **nome dello studente** e **cognome dello studente**. Infine, in basso si prevede di usare una **ListView** per visualizzare i risultati

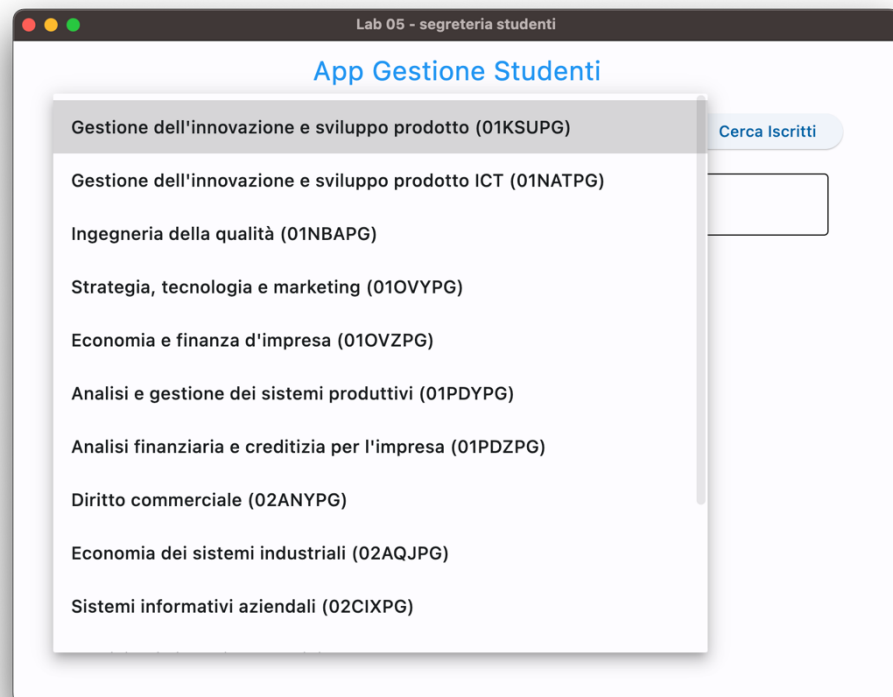
Utilizzare il database *iscritticorsi.sql* (presente nella cartella del progetto), per popolare il menù a tendina con tutti i corsi disponibili.

Nota bene: per popolare il menu a tendina, occorre aggiungere alla sua lista `options` oggetti di tipo `ft.dropdown.Option`. Il costruttore di `ft.dropdown.Option` ammette due parametri:

- `key`: usare il codice `codins`
- `text`: la stringa da visualizzare nel menu a tendina, come nella figura sottostante

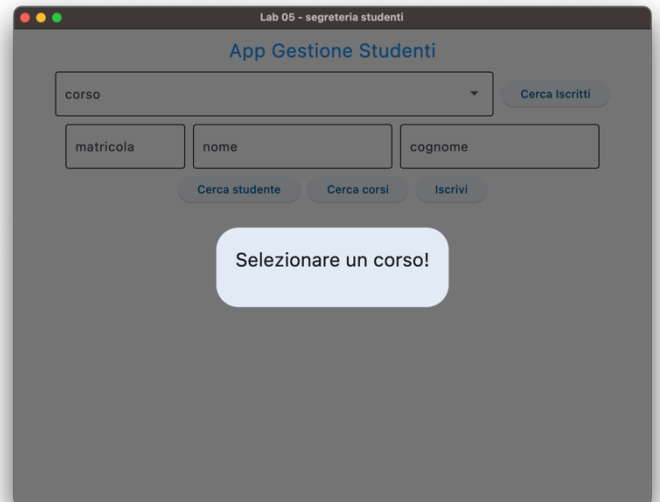
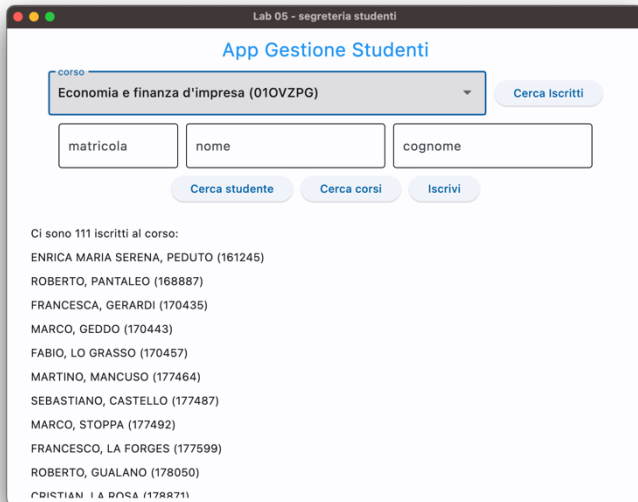
Ovverosia:

```
ft.dropdown.Option (key=corso.codins, text=corso.__str__())
```



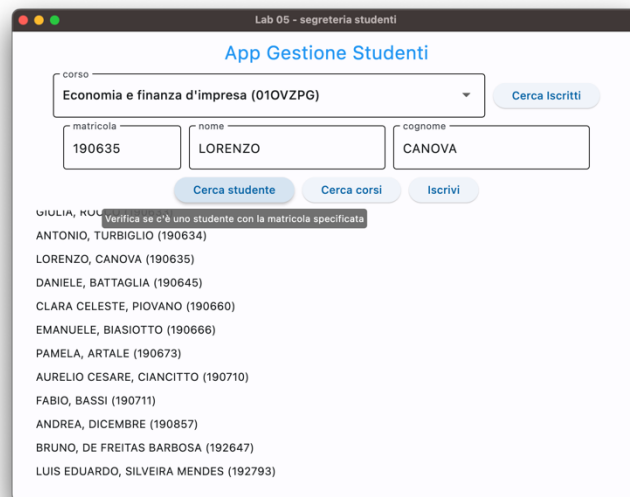
Punto 2

Implementare la ricerca degli studenti iscritti ad un corso: selezionato un corso dal menù a tendina, facendo click sul pulsante *Cerca iscritti*, vengono visualizzati tutti gli studenti iscritti a quel corso. Se nessun corso è selezionato, avvisare l'utente con un messaggio di errore, per esempio scrivendo un messaggio nell'area di testo per i risultati oppure utilizzando un [AlertDialog](https://flet.dev/docs/controls/alertdialog) (<https://flet.dev/docs/controls/alertdialog>)



Punto 3

Implementare la funzionalità di ricerca studente: data la matricola di uno studente, facendo click sul pulsante *Cerca studente*, completare in automatico il suo nome e cognome. Se la matricola non è presente, visualizzare un errore come al Punto 2



Punto 4

Implementare la ricerca dei corsi a cui è iscritto uno studente: data la matricola, facendo click sul pulsante *Cerca corsi*, controllare se lo studente è presente nel database, ed in caso visualizzare tutti i corsi ai quali è iscritto. Se la matricola non è presente, visualizzare un messaggio di errore.

Lab 05 - segreteria studenti

App Gestione Studenti

corso

matricola nome cognome

Risultano 5 corsi:

- Gestione dell'innovazione e sviluppo prodotto ICT (01NATPG)
- Ingegneria della qualità (01NBAPG)
- Economia e finanza d'impresa (01OVZPG)
- Diritto commerciale (02ANYPG)
- Gestione dei progetti (10BDAPG)

Punto 5 (OPZIONALE)

Implementare la funzione di iscrizione di uno studente ad un corso: selezionato un corso ed inserita la matricola di uno studente, facendo click sul pulsante *Iscrivi* è possibile iscrivere lo studente a quel corso.

Per semplicità, questa funzionalità prevede soltanto di iscrivere studenti già presenti nel database in corsi già presenti. Ovvero, si tratta di inserire una nuova relazione nella tabella iscrizioni

