

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-2024/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-2024!), 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 09/10 Aprile 2024

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

The screenshot shows a window titled "Lab 05 - segreteria studenti" containing an application titled "App Gestione Studenti". The interface features a dropdown menu labeled "corso" with the placeholder text "Selezionare un corso". To its right is a text input field labeled "nome corso" and a button labeled "Cerca Iscritti". Below the dropdown and input field are three more text input fields labeled "matricola", "nome", and "cognome". At the bottom of the form are three buttons: "Cerca studente", "Cerca corsi", and "Iscrivi".

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.

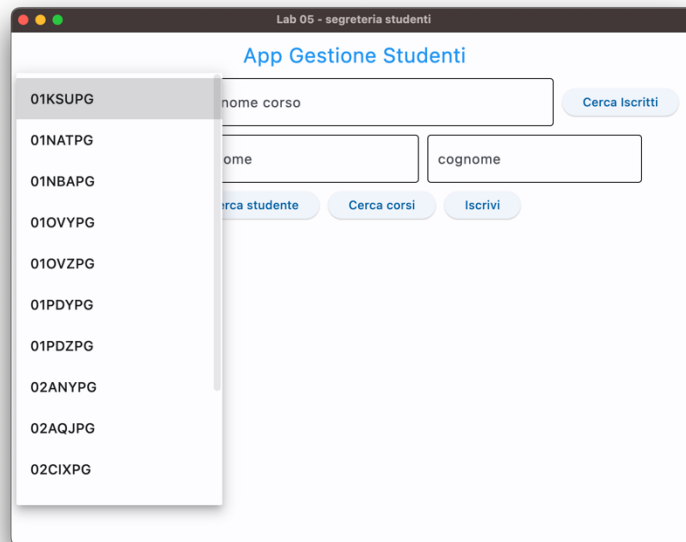
Traccia

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 tre ulteriori campi di testo non-editabili (*read_only=True*) per visualizzare il **nome del corso**, **nome dello studente** e **cognome dello studente**. Infine, in basso si prevede di usare una **ListView** per visualizzare i risultati

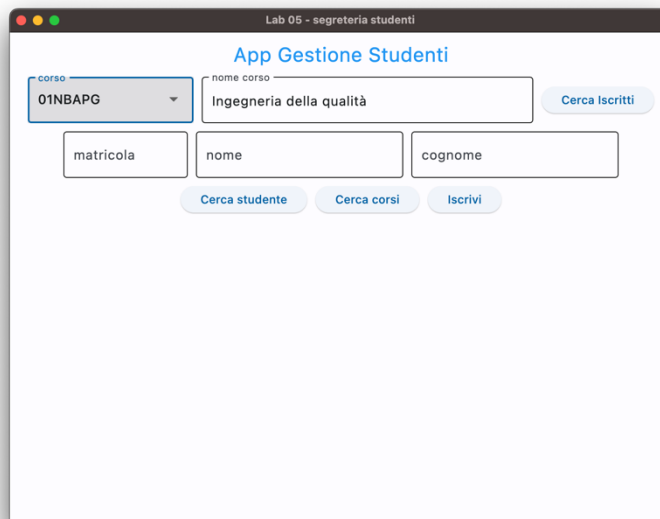
Utilizzare il database *iscrittiscorsi.sql* (presente nella cartella del progetto), per popolare il menù a tendina con tutti i corsi disponibili. Il menu deve visualizzare il codice del corso (codins)

Nota bene: in flet i menu a tendina possono soltanto essere popolati con stringhe, e non con oggetti.



Punto 2

Una volta che viene selezionato dal menu a tendina il codice di un corso, visualizzare nel campo di testo a fianco il nome dell'insegnamento



Punto 3

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>)

The screenshot shows the 'App Gestione Studenti' interface. At the top, there's a title bar 'Lab 05 - segreteria studenti'. Below it, the app title 'App Gestione Studenti' is displayed. The interface includes a dropdown menu for 'corso' with '01NBAPG' selected, a text input for 'nome corso' with 'Ingegneria della qualità', and a 'Cerca Iscritti' button. Below these are input fields for 'matricola', 'nome', and 'cognome'. At the bottom, there are three buttons: 'Cerca studente', 'Cerca corsi', and 'Iscrivi'. The main area displays a list of students enrolled in the selected course:

Ci sono 327 iscritti al corso:
OMAR, PICONE CHIODO (156622)
ENRICA MARIA SERENA, PEDUTO (161245)
FEDERICO, SCALA (161334)
SIMONE, SCHIFANO (167754)
MARINA, BOCCHIARDO (168630)
FRANCESCO, RUSSO (168718)
MANUEL, LORENT (169195)
GIULIANO LEANDRO, PERA (170277)
FRANCESCA, GERARDI (170435)
MARCO, GEDDO (170443)
DANIELE, CONCI (170450)

The screenshot shows the 'App Gestione Studenti' interface with a dark background. A large white dialog box is centered on the screen with the text 'Selezionare un corso!'. The interface elements are dimmed in the background.

Punto 4

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 3

The screenshot shows the 'App Gestione Studenti' interface. The 'corso' dropdown is set to '01NBAPG' and 'nome corso' is 'Ingegneria della qualità'. The 'matricola' field contains '198804', and the 'nome' and 'cognome' fields are filled with 'DANIELE' and 'VASILE' respectively. The 'Cerca Iscritti' button is visible. Below the input fields are three buttons: 'Cerca studente', 'Cerca corsi', and 'Iscrivi'. The main area displays a list of students, with 'DANIELE, VASILE (198804)' at the bottom of the list:

GIAN CARLO, DELL'ERBA (192453)
FRANCESCO, MONTORIO (192686)
RUBEN, CARINI (192978)
CLAUDIA, PARRINELLO (194392)
ANTONIO, CURTO PELLE (196888)
TEODORO, PICCINNI (197220)
SIMONE, BUSON (198030)
ELISABETTA, LISTORTI (198151)
JACOPO, SUPPA (198171)
ANDREA, SCARPELLINO (198175)
FEDERICO, GARINO (198592)
DANIELE, VASILE (198804)

Punto 5

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: nome corso:

matricola: nome: cognome:

Risultano 6 corsi:

- Gestione dell'innovazione e sviluppo prodotto ICT (01NATPG)
- Ingegneria della qualità (01NBAPG)
- Economia dei sistemi industriali (02AQJPG)
- Sistemi informativi aziendali (02CIXPG)
- Analisi dei sistemi economici (02PBVPG)
- Economia aziendale (09AQGGP)

Lab 05 - segreteria studenti

App Gestione Studenti

corso: nome corso:

matricola: nome: cognome:

La matricola indicata non risulta iscritta ad alcun corso

Punto 6 (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

