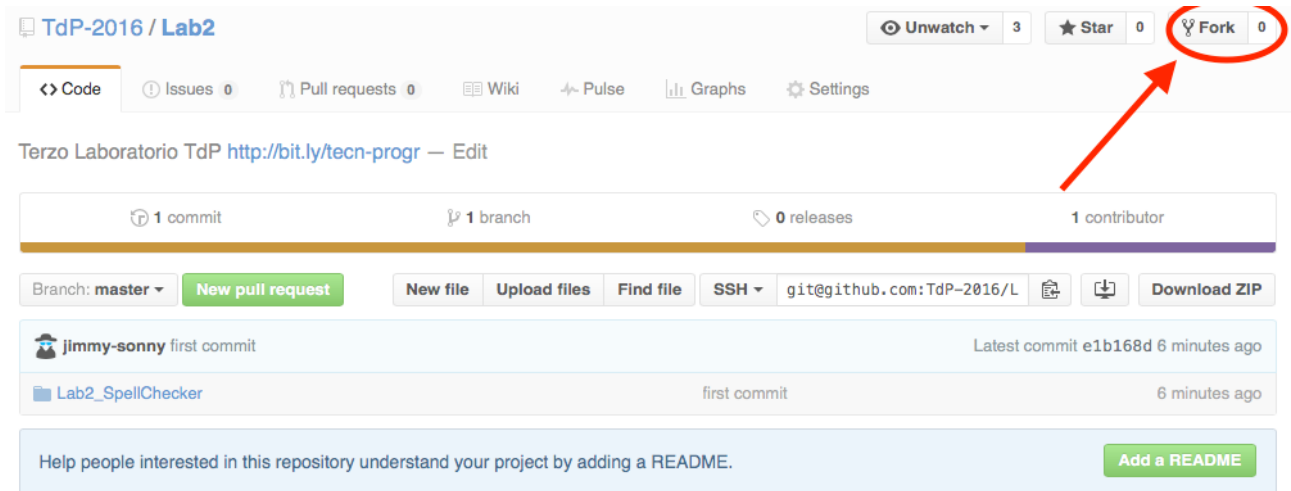


## 03FYZ TECNICHE DI PROGRAMMAZIONE

### 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 terzo laboratorio:  
<https://github.com/TdP-2017/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 Eclipse, andare su *File -> Import*. Digitare *Git* e selezionare *Projects from Git -> Next -> Clone URI -> Next*.
  - Utilizzare la URL del **proprio** repository che si vuole clonare (**non** quello in TdP-2016!), ad esempio:  
<https://github.com/my-github-username/Lab04>
  - Fare click su *Next*. Selezionare il branch (*master* è quello di default) fare click su *Next*.
  - Selezionare la cartella di destinazione (quella proposta va bene), fare click su *Next*.
  - Selezionare *Import existing Eclipse projects*, fare click su *Next* e successivamente su *Finish*.
  - Il nuovo progetto Eclipse è stato clonato ed è possibile iniziare a lavorare.
  - A fine lavoro ricordarsi di effettuare Git commit e push, utilizzando il menù *Team in Eclipse*.

**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 04 – 29 Marzo 2016

---

Obiettivi dell'esercitazione:

- Utilizzo del pattern MVC
  - Introduzione a JDBC
  - Introduzione al Pattern DAO
- 

## ESERCIZIO 1

Dopo aver fatto il fork del progetto relativo al laboratorio 4, realizzare in linguaggio Java un'applicazione dotata di interfaccia grafica che permetta la gestione degli studenti e dei corsi di ateneo. In particolare, l'applicazione permette di *cercare* informazioni relative ad un corso o ad uno studente. Opzionalmente, implementare l'iscrizione di uno studente ad un corso. Fare attenzione alla gestione delle condizioni di errore.

**Punto 1** Realizzare un'interfaccia grafica con *JavaFx* simile a quella mostrata in *Fig 1*. I campi *Nome* e *Cognome* sono dei *TextField* non editabili. Il pulsante *Reset* permette di cancellare il contenuto di tutti i campi.

Utilizzare il database *iscritticorsi.sql* (presente nella cartella del progetto), per popolare il menù a tendina con tutti i corsi disponibili. Inseire anche un campo vuoto da utilizzare quando non si vuole selezionare nessun corso in particolare.

**Fare uso dei pattern MVC e DAO, come spiegato a lezione.**

**Punto 2** Implementare la funzionalità di completamento automatico: data la matricola di uno studente, facendo click sul pulsante verde adiacente, completare in automatico il nome e cognome di uno studente. (Esempio *Fig. 1*)

**Punto 3** Implementare la ricerca degli studenti iscritti ad un corso: selezionato un corso dal menù a tendina, facendo click sul pulsante *Cerca iscritti corso*, vengono visualizzati tutti gli studenti iscritti a quel corso. (Esempio *Fig. 2*). Se nessun corso è selezionato, avvisare l'utente con un messaggio di errore.

**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. (Esempio *Fig. 3*)

**Punto 5** Implementare la funzionalità di ricerca se uno studente è iscritto ad un corso: selezionato un corso ed inserita la matricola di uno studente, facendo click su *Cerca*, è possibile sapere se lo studente specificato è iscritto a quel corso. (Esempio *Fig. 4*)

**Continua nella pagina seguente!**

Controllare di aver gestito tutte le possibili condizioni di errore. (Es. studente non presente, studente non iscritto ad alcun corso, studente già iscritto ad un corso, corso senza iscritti, caratteri non permessi nel campo matricola).

**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. (Esempio *Fig.5*)

Lab3\_Segreteria Studenti

Corso :

Studente:  ✓

**Fig. 1**

Lab3\_Segreteria Studenti

Corso : Analisi dei sistemi economici

Cerca iscritti corso

Studente:   Nome   Cognome

Cerca corsi Iscrivi

|        |            |           |        |
|--------|------------|-----------|--------|
| 146101 | CLAUDIO    | VIGGIANO  | CIN1T3 |
| 148072 | ALBERTO    | VANNINI   | LOP1T4 |
| 154817 | FRANCESCA  | RUSSO     | ORG1T4 |
| 155459 | MARCO      | ROBERTO   | ORG1T4 |
| 155793 | ANDREA     | BARUCCO   | ORG1T4 |
| 156087 | ALESSANDRA | ETERNO    | INF1T3 |
| 158698 | EDOARDO    | MAROLA    | LOP1T4 |
| 166771 | VALERIO    | VIOTTO    | LOP1T4 |
| 170000 | STEFANO    | DOMINICIS | ORG1T4 |

Reset

Fig. 2

Lab3\_Segreteria Studenti

Corso : Corsi

Cerca iscritti corso

Studente:   Nome   Cognome

Cerca corsi Iscrivi

|         |   |   |   |
|---------|---|---|---|
| 01OVYPG | 8 | Strategia, tecnologia e marketing         | 2 |
| 01PDYPG | 8 | Analisi e gestione dei sistemi produttivi | 2 |
| 02ANYPG | 8 | Diritto commerciale                       | 2 |
| 02AQJPG | 8 | Economia dei sistemi industriali          | 2 |
| 02PBVPG | 8 | Analisi dei sistemi economici             | 1 |
| 09AQGPG | 8 | Economia aziendale                        | 1 |

Reset

Fig. 3

Lab3\_Segreteria Studenti

Corso :

Studente:

Fig. 4

Lab3\_Segreteria Studenti

Corso :

Studente:

Fig. 5