

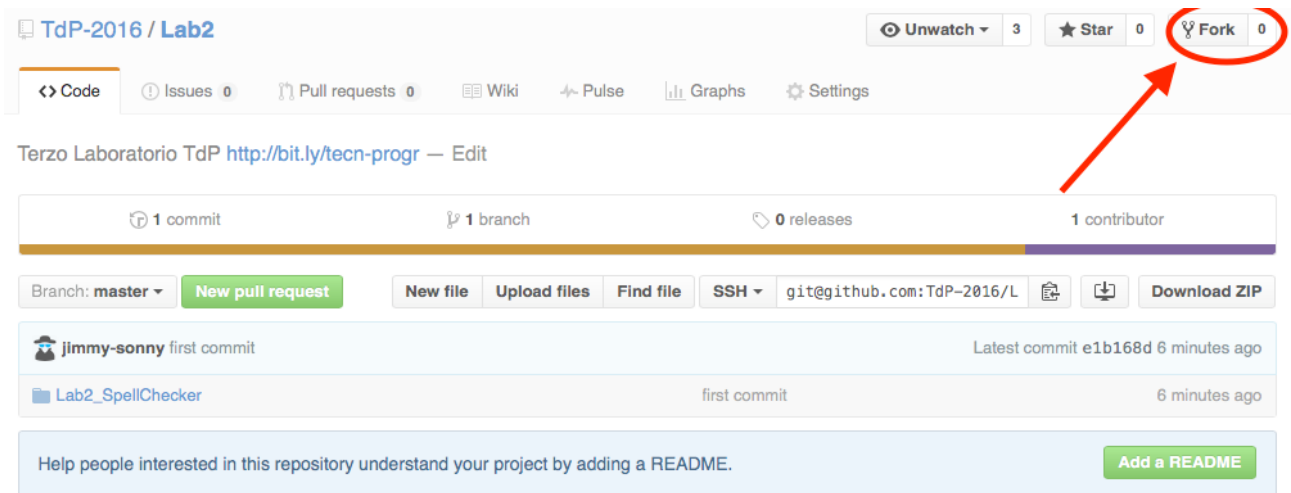
03FYZ TECNICHE DI PROGRAMMAZIONE

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 sesto laboratorio:

<https://github.com/TdP-2017/Lab07>

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-2017!), ad esempio:

<https://github.com/my-github-username/Lab07>

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 07 – 26 aprile 2017

Obiettivi dell'esercitazione:

- Introduzione ai Grafi
 - Utilizzo della libreria JGraphT
-

ESERCIZIO 1

Dopo aver effettuato il *fork* del progetto relativo al settimo laboratorio (Lab07), realizzare in linguaggio Java un'applicazione dotata di interfaccia grafica per rappresentare le somiglianze tra parole della stessa lunghezza nel dizionario Italiano sotto forma di grafo. Date tutte le parole di lunghezza predefinita, ciascuna rappresenta un vertice del grafo. Due parole sono connesse da un arco se differiscono per una sola lettera.

L'applicazione accetta in input il numero di caratteri, pari alla lunghezza delle parole da considerare. Dopo aver cliccato sul bottone “*Genera Grafo*”, utilizzare il database [dizionario.sql](#) per creare un grafo non orientato in cui due parole sono connesse solo se differiscono per una sola lettera. Dopo aver inserito una parola presente nel dizionario e della stessa lunghezza specificata, fare click su “*Trova Vicini*” per visualizzare tutte le parole direttamente connesse a quella inserita. Infine, cliccare su “*Trova tutti i connessi*” per esplorare il grafo seguendo tutte le connessioni possibili a partire dalla parola inserita.

L'applicazione va sviluppata seguendo il pattern MVC e il pattern DAO per l'accesso al database.

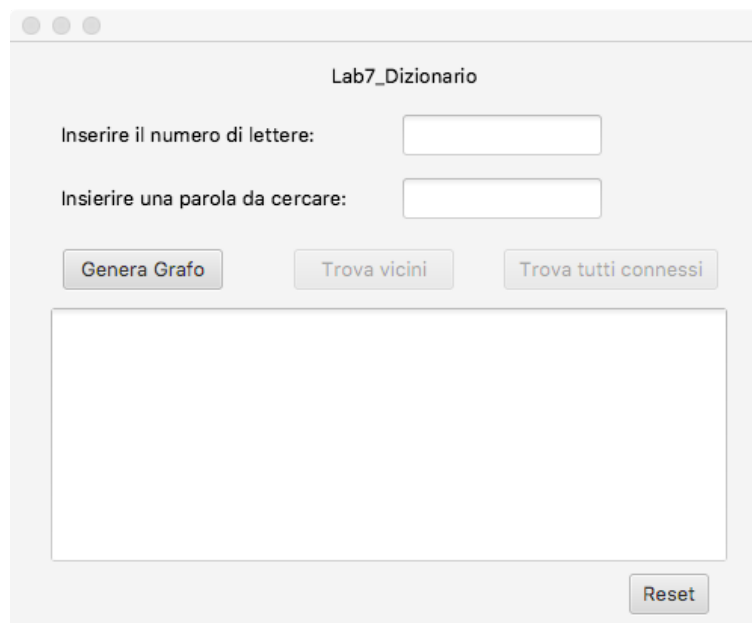


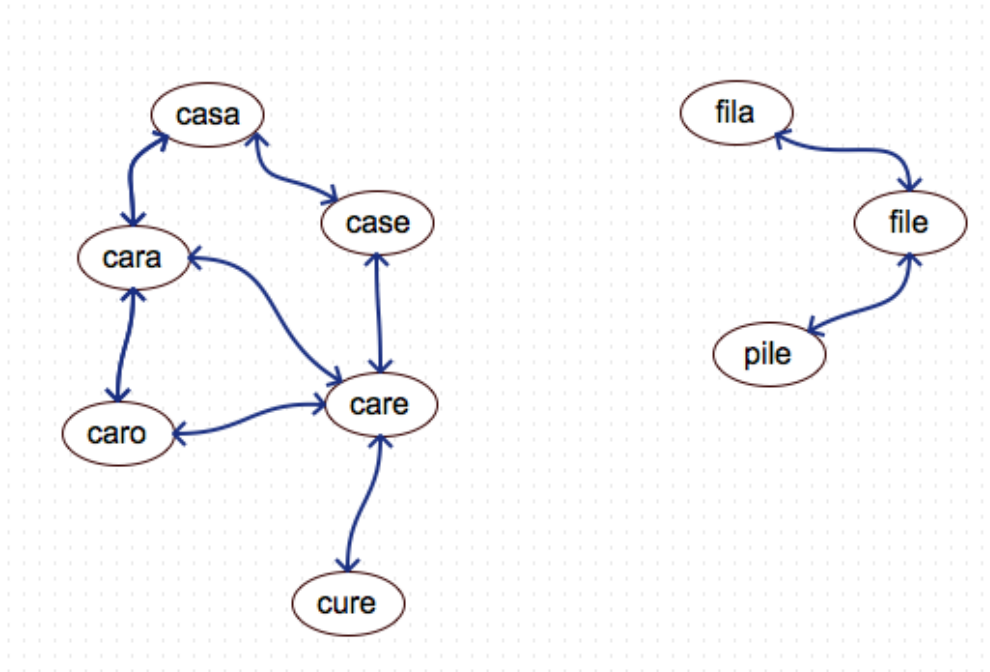
Figura 1: Interfaccia grafica Laboratorio 07

Continua nella pagina seguente

Esempio:

Dato il seguente dizionario di parole di **4 lettere**: [casa, case, cara, care, caro, cure, fila, pila, pile] si ottiene il grafo in Fig 2.

Fig 2



- Inserendo la parola **casa**, si devono ottenere le seguenti due liste:
Vicini: [cara, case]
Tutti i nodi connessi: [cara, case, care, caro, cure]
- Inserendo la parola **fila**, si devono ottenere le seguenti due liste:
Vicini: [pila]
Tutti i nodi connessi: [pila, pile]

Nota: data la dimensione del dizionario si consiglia di utilizzare un numero di lettere piccolo (es. 3, 4, 5).

Nota2: la ricerca delle parole che differiscono di una sola lettera dalla parola data può essere effettuata sia in Java, sia direttamente in SQL. Si consiglia di provare entrambe le soluzioni.

https://www.w3schools.com/sql/trysql.asp?filename=trysql_select_wildcard_underscore

Nota2: la ricerca di tutti i nodi connessi può essere effettuata sia con un algoritmo iterativo o ricorsivo, sia utilizzando la libreria JGraphT. Si consiglia di provare almeno 2 soluzioni alternative.