Relazione progetto   
Legami©

**Studente:** Giorgio Maggiolo

**Matricola:** 610338  
**Anno accademico:** 2010/2011

*Audaces fortuna iuvat*

*(La fortuna aiuta gli audaci)*

*Virgilio, Eneide X, 284*

Sommario

[Introduzione e note di sviluppo 2](#_Toc303958927)

[Variabili di sviluppo 2](#_Toc303958928)

[Note sul progetto ed installazione 2](#_Toc303958929)

[Logica del programma 2](#_Toc303958930)

[Descrizione testuale delle classi: 2](#_Toc303958931)

# Introduzione e note di sviluppo

Legami© è un programma per la gestione di contatti di tipo lavorativo, ovvero un “social network” (con *social network* non viene inteso un sito internet, ma viene presa la sua accezione più generale di “*rete sociale*”)per permettere le interazioni a livello professionale in modo da poter mettere in comunicazione offerta e richiesta di lavoro, oltre che a mettere in contatto enti differenti fra loro.

Questa relazione vuole essere un documento di presentazione del progetto, senza andare troppo nello specifico tecnico, quanto piuttosto descrivere ed esaminare le scelte implementative che sono state attuate.

Ovviamente, però, alcuni dati tecnici dovranno essere dati e per questo motivo sono stati esposti qui di seguito.

## Variabili di sviluppo

* **Sistema operativo:** *Windows 7 Home Edition Service Pack 1 64-bit, Ubuntu 11.04, Ubuntu 10.04*
* **Compilatore:** *MinGw-gcc440-1* e *i486-linux-gnu-4.4.3*
* **Versione libreria Qt**©**:** 4.7.4(32 bit) e 4.6.2

## Note sul progetto ed installazione

La cartella del progetto è divisa in quattro parti:

* La “radice” contiene i file *Legami.pro*, *main.cpp*, e la relazione. Il database, mancante alla consegna, verrà creato automaticamente al primo avvio assieme all’utente amministratore del sistema:
  + Username: root
  + Password: “there is no password” (senza virgolette)
* La cartella “*lib*” contiene i file header e i sorgenti della parte logica del programma, ovvero tutte quelle classi che si occupano della gestione vera e propria delle informazioni e dei dati
* La cartella “*gui*” contiene i file header e i file sorgenti della parte grafica del programma, mera interfaccia per la parte logica
* La cartella “*translation*” contiene i file di traduzione del programma.

Legami© è stato sviluppato nativamente come programma multipiattaforma, per cui non ci sono particolari indicazioni per quanto riguarda la compilazione e l’esecuzione in alcun sistema operativo.

Nel caso dovesse rendersi necessario rigenerare il file di progetto utilizzando il comando “*qmake –project*”, si dovrà aggiungere la stringa “*QT += xml*” in quanto, diversamente, non verrebbero incluse le librerie Qt responsabili della gestione dei file xml e quindi il programma risulterebbe non compilante.

Inoltre la cartella “*translation*” non è necessaria ai fini dell’esecuzione, ma consigliata per godere di una migliore esperienza nell’uso di Legami©.

# Logica del programma

La “logica del programma” è quella parte di Legami deputata alla memorizzazione e all’elaborazione dei dati che vengono immessi dall’utente.

Si trova nella cartella *./lib/* della cartella di progetto ed è strutturata in 13 classi.

L’idea che è stata seguita è divergente rispetto a quanto proposto dal Prof. Ranzato, in quanto non segue minimamente le specifiche minime richieste, anche se riprendono alcuni concetti espressi dal professore nelle specifiche e che sono stati implementati.

Quindi ho reimplementato la struttura di base affinché rispecchiasse la mia idea di questo programma che mi sono fatto basandomi sui già esistenti social network Linkedin e Twitter.

Ma vediamo in generale qual è il compito di queste classi.

## Descrizione testuale delle classi:

* **Legami**: è la madre di tutte le classi logiche, se così possiamo definirla. E’ responsabile di contenere il database di tutti gli utenti, di tutti i gruppi e di tutti i messaggi, oltre a tenere memoria di quale account sia “loggato” in un determinato momento. Contiene diverse funzionalità come la ricerca divisa in varie tipologie, inserimento o rimozione di messaggi, gruppi e utenti. Tiene in memoria anche il puntatore alla classe deputata alla lettura/scrittura nel database.
* **Loader**: questa classe si occupa della lettura e scrittura del database in un file xml. È stata scelta questa implementazione del database perché più semplice rispetto ad un database vero e proprio realizzato in SQLite, in quanto più ampiamente documentato