|  |  |
| --- | --- |
| Gestore richieste di modifica corsi SUPSI e USI | |
|  | |
| Studenti | Relatore |
| Rezart Lohja  Marzio Stampanoni | Riccardo Mazza |
| Correlatore |
| - |
| Committente |
| TI-EDU, eLab USI |
| Corso di laurea | Modulo |
| Ingegneria informatica | Progetto di semestre |
| Anno |  |
| 2017 |  |

|  |  |
| --- | --- |
| Data |  |

**No index entries found.**

Indice delle figure (opzionale, in caso di molte figure)

Indice delle tabelle (opzionale, in caso di molte tabelle)

Indice analitico (opzionale, per parole chiave)

***Abstract***

## 1 Introduzione

## 1.1 Contesto

### *Learning Managment System (LMS)*

Nel mondo l’impiego di infrastrutture digitali per l’aiuto all’apprendimento scolastico é ormai l’abitudine. Nel suo profondo peró un LMS é uno strumento per facilitare ogni aspetto di questo apprendimento attraverso la centralizzazione di tutte le attivitá e risorse su una piattaforma web.

l’utilizzo del framework garantisce un accesso completo a materiale didattico, metodi di valutazione e sistemi per tenere traccia dei propri progressi. Questi software oltre ad essere utilizzati da praticamente ogni universitá hanno svariati utilizzi e sono anche impiegati per organizzare corsi di aggiornamento per aziende e da associazioni sportive per offrire una panoramica globale ai propri utenti dei risultati ottenuti durante le prove.

### *Moodle*

Moodle è un *Learning Managment System* open-source distribuito sotto licenza GNU/GPLv3+ e sviluppato in PHP. Su di esso si basa la piattaforma multimediale *ICorsi* che mette a disposizione di studenti e docenti della Scuola Universitaria Professionale della Svizzera Italiana (SUPSI) e dell’Universitá della Svizzera Italiana (USI) lo spazio dove gestire l’iscrizione degli allievi e lo scambio di dati e materiale relativi ai corsi offerti.

Il framework di sviluppo é estremamente modularizzato e permette cosí l’integrazione di componenti esterne in maniera agevolata

Nel tempo numerose funzionalitá sono state aggiunte sotto forma di plugin sviluppati da studenti durante i loro progetti, dal team di gestione di *ICorsi* - ELearning Lab di USI o acquisiti da internet.

La piattaforma nel suo stato attuale, oltre alla gestione dell’ambiente E-Learning stardard, permette la sincronizzazione automatica dei Database di SUPSI e USI con le tavole di corsi e studenti iscritti all’interno del sistema di Moodle e con il database LDAP esterno necessario ad autenticare gli utenti.

In un corso Moodle agli utenti con privilegi di modifica e concesso organizzare il contenuto didattico secondo i propri bisogni e criteri. Un editor abbastanza intuitivo ma ben documentato (https://docs.moodle.org/32/en/Main\_page, 1 gennaio 2017) permette la creazione di sezioni, l’aggiunta di documenti, la richiesta di consegna di un esercizio e numerose altre *Attivita* e *Risorse*. Un pannello *Administration* offre invece la possibilitá di iscrivere utenti al corso ed infine la gestione dei voti e dei settaggi globali del corso.

# Descrizione del lavoro

## Situazione attuale

La struttura di ruoli e permessi di Moodle, che verrá descritta piú avanti nel presente documento, isola a livelli di accesso le modifiche che il singolo utente puó effettuare. Per questo motivo e per decisioni strutturali di implementazione del framework alla base la creazione di un corso e l’iscrizione dei partecipanti all’interno di Moodle deve essere eseguita da un utente con livello di accessibilitá *Manager* o *Admin* e non é automatizzata.

Un form interno permette agli utenti con capacitá di modifica l’inoltro di richieste per la creazione o cancellazione di un corso ma per motivi pratici e per la natura spesso complicata di tali richieste questo sistema non viene utilizzato presso USI e SUPSI. Docenti e assistenti che intendono aprire un nuovo corso devono inoltrare una richiesta attraverso un canale esterno e non hanno la garanzia di specificare in maniera efficace i dati necessari.

La mancanza di un interfacciamento tra il database di Moodle e le banche dati interne delle due scuole ticinesi non permette inoltre una visione dei dati necessari alla creazione dei corsi all’interno della piattaforma, il luogo dove verranno poi svolte le azioni di modifica.

## Progetto

L’obiettivo di questo progetto è quello di permettere di inoltrare, modificare e gestire le richieste per la creazione e la cancellazione di corsi direttamente da Moodle tramite lo sviluppo di un plugin ad-hoc che sia in grado di comunicare con i database esterni di USI e SUPSI per poter acquisire le informazioni necessarie e tuttora mancanti.

Sono state individuate dal committente tre tipologie di richiesta che docenti e assistenti devono poter effettuare. Inoltre sia il manager di categoria che chi ha inoltrato la richiesta devono poter gestire quest’ultima seguendola nel suo ciclo di vita e nel caso modificandone i dati principali ed eventuali persone iscritte al corso.

Gli utenti devono poter:

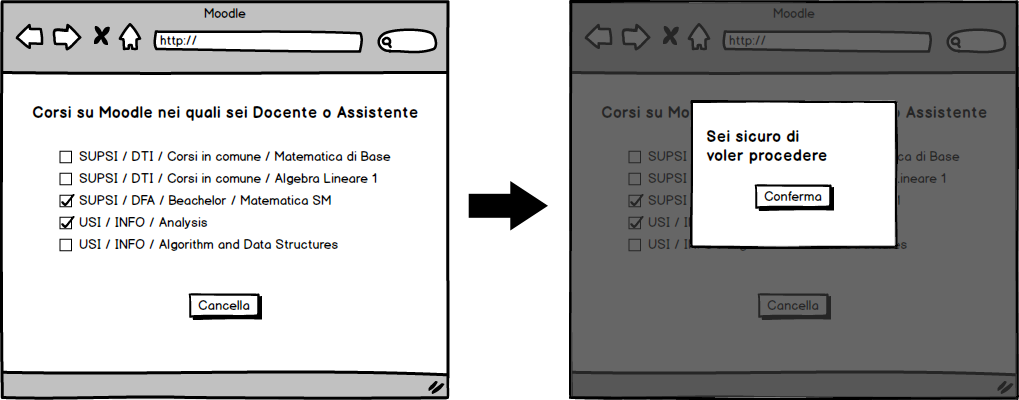
1. **Cancellare** un corso esistente – All’utente va presentata una vista di tutti i corsi nei quali è personalmente coinvolto con il ruolo di *Docente* o *Assistente* sul database di Moodle. Dopo avero fatto la propria selezione un messaggio di allerta deve chiedere conferma per effettuare l’inoltro della cancellazione.
2. **Clonare** un corso esistente – Come per la cancellazione di un corso all’utente va presentata una vista di tutti i corsi nei quali è personalmente coinvolto prendendo i risultati dal database di Moodle. Dopo la scelta una vista di *Resume* deve permettere di cambiare i dati essenziali del corso da clonare come *titolo*, *nome breve*, *categoria*, *docenti iscritti* e *assistenti iscritti.*
3. **Creare** un corso – In una prima fase l’utente puó scegliere tra i corsi presenti sul database di SUPSI e USI ma non sono ancora presenti all’interno di ICorsi. Va fatta distinzione tra i corsi con il quale l’utente ha una relazione con il proprio ruolo e gli altri ed é inoltre richiesto un filtro a stringhe per facilitare l’utente nella ricerca. Secondariamente l’utente deve poter creare un corso senza partire da nessun dato iniziale, specificando quindi tutti i parametri necessari per la succesiva creazione da parte di un *manager*. Anche questa tipologia di richiesta termina in una vista di *Resume* per le ultime modifiche e la conferma finale
4. **Gestire** le richieste
   1. User – Ad un utente con ruolo docente o assistente viene garantita la gestione delle proprie richiete effetuate. Ha la possibilitá di modificarne i dati e salvarne i cambiamenti oppure di cancellarla se lo ritiene necessario.
   2. Manager – Ad un utente con ruolo di manager o admin viene invece dato accesso alla gestione dei dati e dello stato delle richieste di tutti gli utenti.

## Requisiti

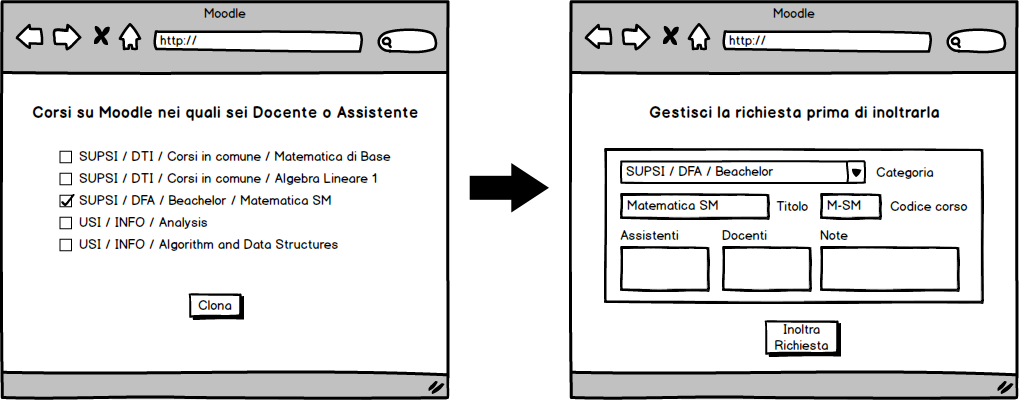
Il risultato finale deve garantire un plugin installabile sulla versione scaricabile di Moodle al momento dell’esecuzione del progetto. e dei *Web Services* per interfacciarsi con i database di USI e SUPSI per accedere alle informazione dei corsi. Di seguito sono elencati i requisiti fondamentali del progetto.

Il software sviluppato dovra implementare le quattro viste elencate al punto 2.2 con tutte le funzionalita per la gestione e l’inoltro delle richieste. Per meglio comprendere quanto ci é stato richiesto di sviluppare analizziamo come l’utente dovrá poter eseguire le azioni durante la navigazione. Per capire il *flow* di esecuzione e di conseguenza le tecnologie da implementare per farle funzionare correttamente suddividiamo di nuovo questa analisi nelle quattro principali azioni coinvolte.

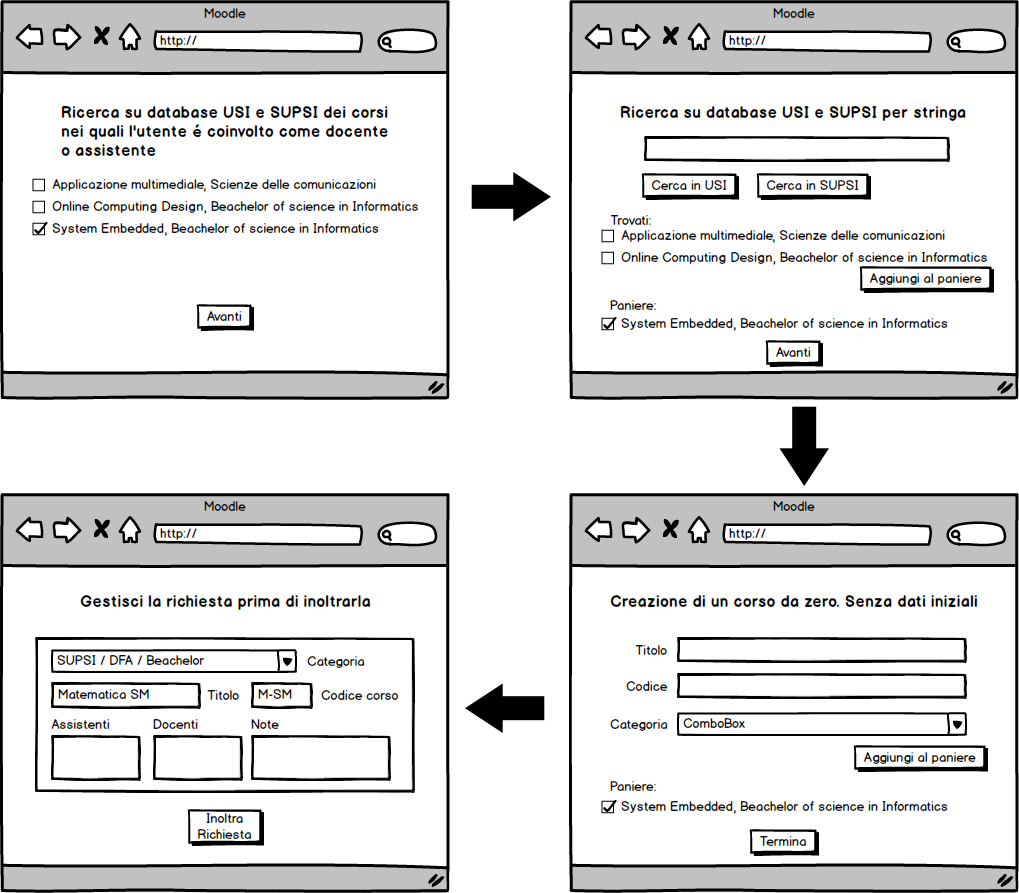
1. **Cancellazione**



1. **Clonazione**

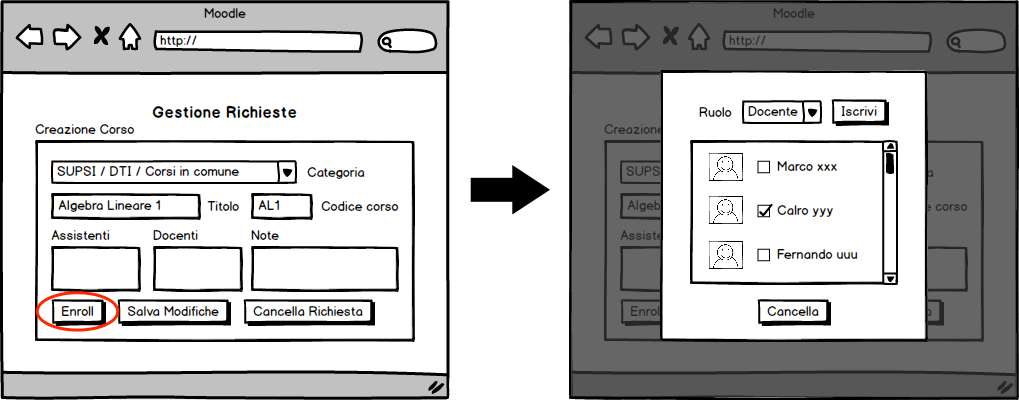


1. **Creazione**



1. **Manage**

Anche se le funzionalitá proposte in questa parte sono diverse a dipendenza dei privilegi dell’utente che sta consultando la pagina, le meccaniche di utilizzo sono le medesime.



La lettura e scrittura dai database di USI e SUPSI da Moodle comporta l’utilizzo di Web Services appositi che tramite un’interfaccia di comunicazione di tipo REST sono in grado di gestire richieste e comunicare con l’esterno. Vanno quindi implementate le classi per comporre e fare le richieste.

Ovviamente non ci é possibile eseguire *que*ry direttamente sui database delle universitá e non esiste ancora un API che espone la possibilitá di accedere a queste risorse. Ci é stato peró assicurato che la parte di processamento delle richieste e la preparazione dei dati verrá sviluppata in un secondo momento. Il nostro compito é quello di fornire una volta terminato il progetto una guida su quali dati servono e la loro struttura in modo da rendere il plugin indipendente da chi fornisce i dati. Per il nostro tempo di sviluppo dobbiamo organizzarci una classe di *mock-up* che risponda alle nostre chiamate come se fosse effettivamente una banca dati con un Database Managment System (DBMS) che la gestisce.

Per garantire la buona riuscita del progetto e realizzare quanto richiesto é necessario lo studio del funzionamento di Moodle e di come va implementato un plugin per estenderlo. Va inoltre preparato un ambiente di lavoro con un server per PHP e MySQL sul quale installare il Framework.

Come “*Nice To Have”* viene richiesto di indagare sulla possibilitá di automatizzare la creazione di un corso ed eventualmente aggiungere questa funzionalitá nella parte di *manage* delle richieste. I privilegi di compiere cancellazioni e creazioni dei corsi vanno ovviamente offerti solamente ad utenti con ruoli specifici.

## 3. Architettura

## 3.1. Database

l’implementazione della logica del plugin e quindi tenere traccia delle richieste da gestire con tutti i dati correlati ha richiesto l’aggiunta di tre tavole nel database di Moodle. Ad ogni richiesta sono associati da uno a infiniti corsi e rispettivamente ogni corso puo avere da zero ad infiniti utenti. In queste tavole sono salvate tutte le informazioni necessarie alla futura creazione o cancellazione di un corso per ogni richiesta effettuata.

In *lcl\_courseseditor\_richieste* sono immagazzinati solo i dati relativi all’utente e alla data di inoltro della richiesta tutto quanto appartiene ad ogni singola di esse viene invece salvato nelle tabelle *lcl\_couseseditor\_corso* nella quale é presente una *foreign key* per relazione verso la richiesta.

**lcl\_courseseditor\_richiesta**

|  |  |  |
| --- | --- | --- |
| **#** | **Name** | **Type** |
| 1 | id (chiave primaria) | bigint |
| 2 | id\_mdl\_user | bigint |
| 3 | data\_richiesta | bigint |

La tabella *corso* oltre a descrivere le caratteristiche principali per un corso Moodle si occupa anche di salvare alcune caratteristiche delle richieste. Siccome ogni richiesta puó essere composta da piú di un corso spetta quindi ad ognuno di essi conoscere lo stato della richiesta e il tipo.

**lcl\_courseseditor\_corso**

|  |  |  |
| --- | --- | --- |
| **#** | **Name** | **Type** |
| 1 | id (chiave primaria) | bigint |
| 2 | id\_lcl\_courseseditor\_richiesta | bigint |
| 3 | titolo | longtext |
| 4 | shortname | varchar |
| 5 | id\_mdl\_course\_categories | bigint |
| 6 | id\_mdl\_course | bigint |
| 7 | note | longtext |
| 8 | tipo\_richiesta | varchar |
| 9 | stato\_richiesta | varchar |

Ad ogni corso possono essere associati degli utenti assegnati come docenti o assistenti. Il campo *tipo\_relazione* sta a specificare quale ruolo ha lo user all’interno del corso al quale va iscritto. Valori come questo ma anche stato e tipo nella tabella precedente sono passati sotto forma di costanti tramite PHP e risultano avere il ruolo di *flag*.

**lcl\_courseseditor\_user\_corso**

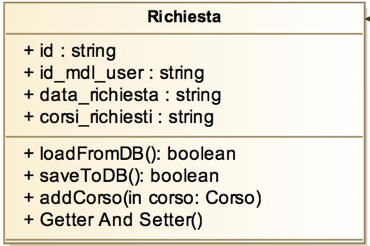
|  |  |  |
| --- | --- | --- |
| **#** | **Name** | **Type** |
| 1 | id | bigint |
| 2 | id\_lcl\_courseseditor\_corso | bigint |
| 3 | tipo\_relazione | varchar |
| 4 | nome | varchar |
| 5 | cognome | varchar |
| 6 | id\_mdl\_user | bigint |

## 3.1.1. Schema delle relazioni

## 3.2. Classi di mapping con database Moodle

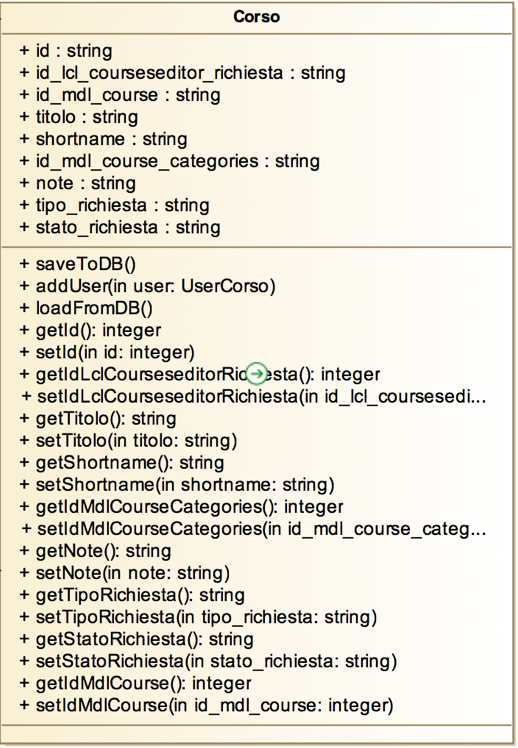
Per la gestione corretta ed efficace delle tavole elencate nei punti precedenti sono state create tre rispettive classi in PHP con la funzione di offrire un mapping fra le righe in database e oggetti da poter utilizzare e manipolare dentro al programma. Per ottenere questo scopo ogni classe contiene un *field* per ogni rispettivo campo presente in database. Tutti gli oggetti espongono poi i metodi *loadFromDb()* e *saveToDb().* abbiamo quindi creato un interfaccia che tutte le classi implementano. I due metodi permettono di caricare e salvare i dati necessari per la gestione delle richieste. Gli oggetti con i dati espongono infine i getter e i setter per poter accedere e modificare tali valori.

1. **Richieste** – La classe Richiesta funge da interfaccia con la tabella *mdl\_lcl\_courseseditor\_richiesta*. Espone il metodo *addCorso(Corso)* che permette di aggiungere un corso all’*arrayList*. Ogni richiesta infatti puó contenere piú di un corso su cui è stata pretesa un azione. Vi é infine l’id dell’utente che ha effettuato tale richiesta e la data del suo inoltro / ultima modifica.

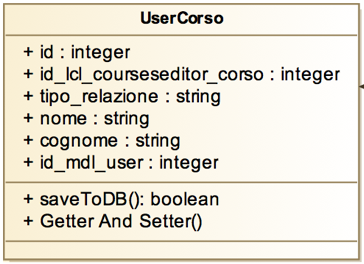


1. **Corso** – Corso fa da interfaccia con la tavola *mdl\_lcl\_coursesditor\_corso*. Per una questione di consistenza nei dati ogni corso é ha conoscenza del proprio *ID moodle* e della categoria a cui appartiente all’interno della piattaforma. Oltre a questi due valori i campi della classe essenziali per la succesiva creazione del corso sono *title* e *shortname.*

La relazione *uno a n* con la classe Richiesta impone ad ogni corso l’obbligo di conoscenza dell’id della richiesta a cui appartiene. Il metodo *addUser(UserCorso)* fornisce la possibilitá di iscrivere un utente al corso in creazione.

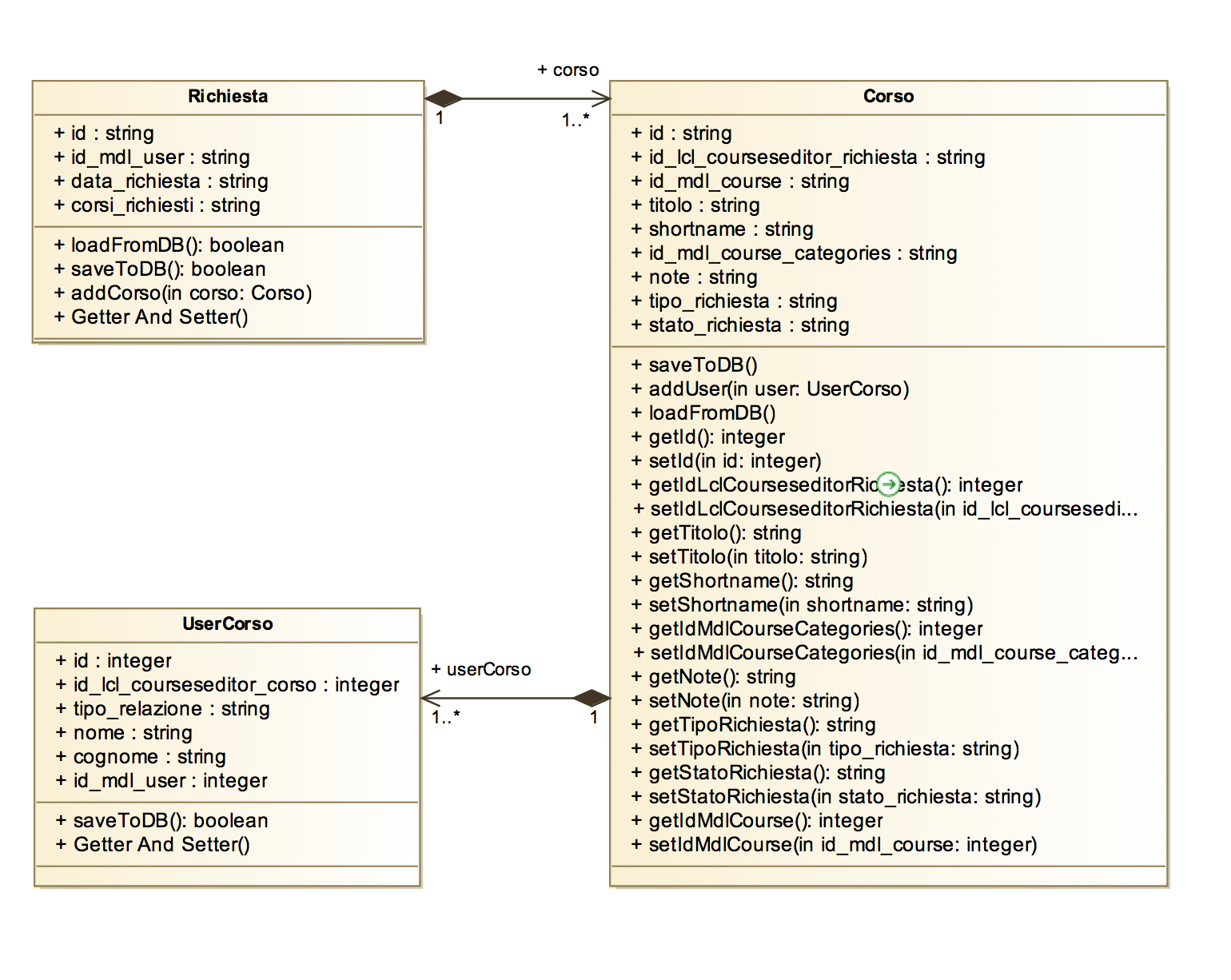


1. **UserCorso** – Questa classe si occupa di fare da interfaccia per la tabella *mdl\_lcl\_courseseditor\_usercorso.* Anche in questo caso vi é un mapping tra gli utenti in questa tabella e i corrispettivi utenti presenti nella tabella *mdl\_user* contentente i dati relativi a tutta la piattaforma. Il campo *id\_mdl\_user* si occupa di immagazzinare tale informazione.



## 3.2.1 Relazioni tra le classi di gestione DB

Per mantenere coerenza con le relazioni delle tavole a cui fanno riferimento in banca dati le classi seguono il medesimo schema di relazioni. Esso viene raffigurato nell’immagine seguente. Tutte le classi implementano inoltre l’interfaccia comune CESSO e sono quindi forzate ad implementarne i metodi.

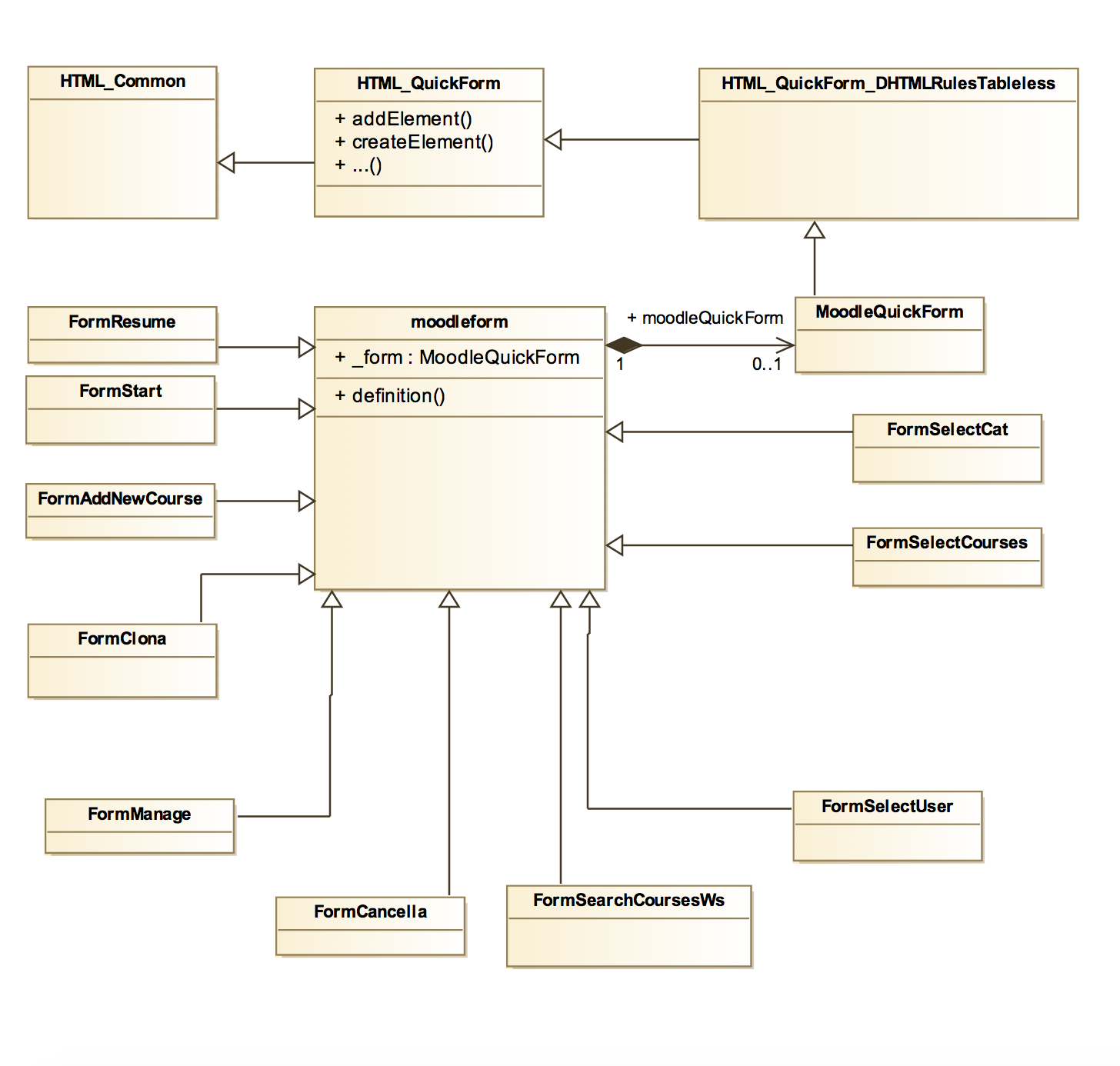


Quando avviene una chiamata al metodo *saveToDb()* di *Richiesta,* dopo aver salvato nella tavola *lcl\_courseseditor\_richieste* una nuova riga con i dati di creazione visti prima, viene in effetti chiamato lo stesso metodo su tutti i corsi che appartengono alla richiesta. A catena ogni corso si occupa di scorrere la lista di utenti iscritti e ancora una volta chiamare *saveToDb()* su ognunodi essi. Lo stesso ragionamento vale per *loadFromDb()*. In entrambi i casi comunque é garantito un’interfacciamento pratico con le tavole del database, alla necessitá di ottenere tutti i dati di un corso una singola chiamata ci permette di ottenere anche i dati degli utenti apparteneti a tale corso. Questa scelta ha snellito di parecchio le operazioni di salvataggio e caricamento da database Moodle.

## 3.3. Classi di collegamento con i web services

# 

## 3.3. Classi di gestione dei form



2.1.2

* **intercambiabilità**: Lorem ipsum dolor sit amet, consectetur adipiscing elit.
* **intercambiabilità**: Lorem ipsum dolor sit amet, consectetur adipiscing elit.
* **intercambiabilità**: Lorem ipsum dolor sit amet, consectetur adipiscing elit.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut rutrum fringilla tortor, id ullamcorper mauris elementum sit amet. Mauris nec tellus purus;

1. Lorem ipsum dolor sit amet, consectetur adipiscing elit.;
2. Lorem ipsum dolor sit amet, consectetur adipiscing elit..

## 1.2 XXXXXXXXXXXXC

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut rutrum fringilla tortor, id ullamcorper mauris elementum sit amet. Mauris nec tellus purus,

### 1.2.1 XXXXXXXXXXXXXX

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut rutrum fringilla tortor, id ullamcorper mauris elementum sit amet. Mauris nec tellus purus,

**Capitolo 2**

# 2 XXXXXXXXXXXXXXX

## 2.1. XXXXXXXXXXXX

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut rutrum fringilla tortor, id ullamcorper mauris elementum sit amet. Mauris nec tellus purus, vitae sollicitudin nunc. Aenean suscipit, ipsum vel vulputate lobortis, orci purus consectetur felis, sed luctus neque dui eu odio. Aliquam risus nisl, egestas eu accumsan non, luctus in nibh. Nulla consectetur lacus eu enim imperdiet rhoncus. Proin lobortis tincidunt eros eu feugiat.

Tabella: 0‑1: Applicazione tipiche del disegno a mano libera:

|  |  |  |  |
| --- | --- | --- | --- |
| **Evento** | **Esempio** | **Obiettivo** | **Criteri** |
| XXXXXXXXX | XXXXXXXXX | XXXXXXXXX | XXXXXXXXX |
| XXXXXXXXX | XXXXXXXXX | XXXXXXXXX | XXXXXXXXX |
| XXXXXXXXX | XXXXXXXXX | XXXXXXXXX | XXXXXXXXX |
| XXXXXXXXX | XXXXXXXXX | XXXXXXXXX | XXXXXXXXX |

## 2.2 XXXXXXXXXXXX

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut rutrum fringilla tortor, id ullamcorper mauris elementum sit amet.

Figura 0‑1: XXXXXXXXXXX

### 2.2.1 XXXXXXXXXXXXXX

Generalmente le 6 viste permettono di rappresentare in maniera esaustiva la forma esterna di praticamente tutti gli oggetti. Di regola il particolare viene disegnato nelle viste sufficienti per una sua rappresentazione esaustiva.

**Capitolo 3**

# 3 XXXXXXXXXXXXXXX

## 3.1 XXXXXXXXXXXX

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut rutrum fringilla tortor, id ullamcorper mauris elementum sit amet. Mauris nec tellus purus, vitae sollicitudin nunc. Aenean suscipit, ipsum vel vulputate lobortis, orci purus consectetur felis, sed luctus neque dui eu odio. Aliquam risus nisl, egestas eu accumsan non, luctus in nibh. Nulla consectetur lacus eu enim imperdiet rhoncus. Proin lobortis tincidunt eros eu feugiat.

Tabella: 0‑1: Applicazione tipiche del disegno a mano libera:

|  |  |  |  |
| --- | --- | --- | --- |
| **Evento** | **Esempio** | **Obiettivo** | **Criteri** |
| XXXXXXXXX | XXXXXXXXX | XXXXXXXXX | XXXXXXXXX |
| XXXXXXXXX | XXXXXXXXX | XXXXXXXXX | XXXXXXXXX |
| XXXXXXXXX | XXXXXXXXX | XXXXXXXXX | XXXXXXXXX |
| XXXXXXXXX | XXXXXXXXX | XXXXXXXXX | XXXXXXXXX |

## 3.2 XXXXXXXXXXXX

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut rutrum fringilla tortor, id ullamcorper mauris elementum sit amet.

I capitoli numerati contengono la parte principale del documento e sono in genere suddivisi in sottocapitoli ed eventualmente in ulteriori sottocapitoli (max. 3 livelli),

Ulteriori piccole suddivisioni che si rendessero necessarie possono essere ottenute, senza aumentare il numero di livelli, con dei titoli intermedi senza numerazione o con trattini / palline iniziali.

Ogni capitolo risp. sottocapitolo deve avere un contenuto di una certa entità, cioè non consistere soltanto di qualche riga o di una figura, ma raccogliere testi e figure che si riferiscono a uno stesso argomento di una certa ampiezza.

Quindi per capitoli brevi, ad esempio per l’introduzione, può non essere opportuna una suddivisione in sottocapitoli.

Figure e tabelle vanno sempre referenziate nel testo principale (la prima volta prima dell’inserimento della figura/tabella stessa) e non devono costituire sottocapitoli a sè.

Per figure, grafici, diagrammi, schemi vari la didascalia (spiegazione) va posta al di sotto, nella forma “Figura C.N. Testo”, dove C = num. capitolo, N = num. progressivo all’interno del capitolo. Non si scende normalmente a livello di sottocapitolo, a meno che ci sia un numero molto elevato di figure/tabelle.

Es.: Figura 2.3. Schema di principio del circuito.

Per le tabelle invece la didascalia va posta al di sopra, cioè prima della tabella, in forma analoga a quella delle figure.

Es.: Tabella 1.3. Valori di potenza in funzione del carico.

Le formule che vengono referenziate nel testo vanno identificate con un riferimento numerico progressivo a destra della formula stessa e analogo a quello per le figure e tabelle, nella forma C.N, racchiuso tra parentesi tonde.

Es.: formula: a+b+c = d (2.3),

riferimento: “… come risulta dalla formula (2.3) …”

Piani di Lavoro

Qui va inserita la pianificazione dei lavori di progetto, con almeno il piano iniziale previsto e quello finale di realizzazione effettiva.

Se necessario o utile possono essere inseriti anche dei piani intermedi.

Bibliografia

Va suddivisa secondo le fonti e con riferimenti numerici tra parentesi quadre [n], che vanno inseriti nel testo subito dopo la parola o frase cui quel riferimento bibliografico si riferisce.

Indicazioni di libri:

[n] autore/autori, titolo del libro, editore, anno, edizione

Indicazioni di articoli:

[n] autore/autori, titolo dell’articolo, titolo della rivista, volume, data, pagine specifiche

Indicazioni di riferimenti Internet:

[n] indirizzo Internet, event. pagine specifiche da selezionare

Appendici

Sono inserite direttamente nel fascicolo della documentazione.

Vanno identificate singolarmente con A1, A2, A3, ecc. e i relativi titoli, oppure a gruppi con A1, A2, ecc., B1, B2, ecc., C1, C2, ecc., nel caso si vogliano evidenziare dei blocchi di appendici dello stesso tipo, come ad es. schemi, diagrammi, listati, ecc.

Nel caso di appendici su cui sia impossibile o troppo complicato inserire la numerazione delle pagine, come ad esempio listati già formattati, va inserita una prima pagina numerata che riporti il contenuto e il numero di pagine dell’appendice stessa, facendo poi seguire le pagine originali (con numerazione autonoma o senza numerazione).

Allegati

Fanno parte della documentazione, ma non del relativo fascicolo, trattandosi di materiale separato, anche se riferito alla documentazione stessa.

Si tratta del CD contenente la documentazione stessa e altro materiale riferito al progetto, ev. fascicoli separati (come ad es. un Manuale d’uso), ev. materiale sperimentale del progetto.

L’identificazione avviene come per le Appendici, usando però lettere iniziali diverse, in modo da non confondere Appendici e Allegati.

Importante: gli Allegati, essendo separati, vanno sempre corredati di scritte o etichette che li identifichino come relativi al progetto e alla documentazione cui si riferiscono (titolo, codice, ecc.).

Titolazione > (testo Arial 18)

Sottotitoli e paragrafi introduttivi > (testo Arial 11)

The lazy duck > (Arial bold 9)

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut rutrum fringilla tortor, id ullamcorper mauris elementum sit amet. Mauris nec tellus purus, vitae sollicitudin nunc. Aenean suscipit, ipsum vel vulputate lobortis, orci purus consectetur felis, sed luctus neque dui eu odio. Aliquam risus nisl, egestas eu accumsan non, luctus in nibh. Nulla consectetur lacus eu enim imperdiet rhoncus. Proin lobortis tincidunt eros eu feugiat. Duis erat tellus, imperdiet vel luctus, lobortis eget lacus. In nunc nisl, venenatis euismod dapibus sed, fringilla ac velit. Cras in augue nibh, vitae placerat nunc. Duis vitae sapien a eros vestibulum sollicitudin. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Praesent non tristique mi. Mauris a libero vel velit feugiat fermentum in in ligula. Aliquam sodales ipsum vel odio auctor dictum. > (testo Arial 9)

* Mauris et lacus turpis, a elementum elit. Vestibulum pharetra viverra libero, eu scelerisque ante placerat a.
* Nunc ac neque sit amet nibh dictum tincidunt et sit amet nulla.
* Fusce feugiat purus sit amet magna ullamcorper et accumsan magna venenatis. Pellentesque lectus orci, congue et iaculis et, sollicitudin ac massa.
* Morbi mattis ligula et ligula scelerisque eu convallis orci consectetur. Nullam ultrices dictum dolor, in aliquet turpis scelerisque et. Proin fermentum quam nec neque aliquam non euismod ipsum facilisis. Vivamus elementum vehicula enim, at bibendum quam rutrum sit amet. Nulla facilisi.
* In hac habitasse platea dictumst. Aliquam eget ante massa. Cras vitae nunc ac mi ultrices tristique vitae et nulla. Praesent ut mauris dolor, non placerat nibh.
* Duis dignissim quam vitae erat elementum non consequat arcu tristique. Sed eget sem augue. Donec nulla orci, sodales varius tempor ac, iaculis in mauris. > (elenco puntato Arial 9)

The lazy lizard > (titoletto Arial bold 10)

Proin fermentum quam nec neque aliquam non euismod ipsum facilisis. Vivamus elementum vehicula enim, at bibendum quam rutrum sit amet. Nulla facilisi. In hac habitasse platea dictumst. Aliquam eget ante massa. Cras vitae nunc ac mi ultrices tristique vitae et nulla. Praesent ut mauris dolor, non placerat nibh. Duis dignissim quam vitae erat elementum non consequat arcu tristique. Sed eget sem augue. Donec nulla orci, sodales varius tempor ac, iaculis in mauris. > (testo Arial 10)

* Nam euismod aliquet odio, ac egestas ante luctus quis. Pellentesque et lorem lorem, eget vehicula mi. In vulputate massa quis diam feugiat id consectetur risus pretium. Vestibulum nisi magna, mollis ut dignissim et, porttitor non leo. Vestibulum quis sem sed enim tempor rutrum. Donec luctus tempus metus, congue sodales orci ullamcorper a. Vestibulum nec bibendum lorem. Mauris nec sem ut tellus tincidunt congue sed in nunc. Curabitur pretium, justo non congue lacinia, quam mi faucibus mi, eget viverra risus orci at eros. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Aliquam bibendum justo convallis erat tincidunt lacinia. Nunc viverra rutrum augue nec posuere. Curabitur euismod massa nec elit euismod in rhoncus neque viverra. > (elenco puntato Arial 10)

The lazy dog > (titoletto Arial bold 11)

In sem sapien, posuere non eleifend id, porta et augue. Praesent ultricies suscipit magna, ut volutpat leo mollis eu. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Ut sit amet felis eleifend massa blandit dictum id vel diam. Proin dapibus sapien nec urna venenatis adipiscing. Sed tincidunt congue eros sit amet aliquet. Donec ultricies bibendum justo, viverra dignissim metus fringilla ac. Etiam odio eros, luctus vel bibendum a, pretium non dui. Quisque ac nibh vitae lorem mollis tempus. Quisque blandit mattis purus, tempor vehicula orci congue a. Vestibulum scelerisque commodo sem volutpat eleifend. Integer in porttitor sapien. Nullam accumsan faucibus quam eu tristique. Maecenas nibh justo, vehicula et gravida id, placerat in erat. Cras et mauris lacus. Aenean nulla ligula, venenatis consequat convallis ut, imperdiet vel mauris. In lobortis mattis dolor, ac lobortis orci feugiat ut. Ut vestibulum nisl sed risus vestibulum vehicula. > (testo Arial 11)

* In sem sapien, posuere non eleifend id, porta et augue. Praesent ultricies suscipit magna, ut volutpat leo mollis eu. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Ut sit amet felis eleifend massa blandit dictum id vel diam. Proin dapibus sapien nec urna venenatis adipiscing. Sed tincidunt congue eros sit amet aliquet. Donec ultricies bibendum justo, viverra dignissim metus fringilla ac. Etiam odio eros, luctus vel bibendum a, pretium non dui. Quisque ac nibh vitae lorem mollis tempus. Quisque blandit mattis purus, tempor vehicula orci congue a. Vestibulum scelerisque commodo sem volutpat eleifend. Integer in porttitor sapien. Nullam accumsan faucibus quam eu tristique. Maecenas nibh justo, vehicula et gravida id, placerat in erat. Cras et mauris lacus. Aenean nulla ligula, venenatis consequat convallis ut, imperdiet vel mauris. In lobortis mattis dolor, ac lobortis orci feugiat ut. Ut vestibulum nisl sed risus vestibulum vehicula. > (elenco puntato Arial 11)

Nome Cognome (1. Firma) Nome Cognome (2. Firma)

Funzione Funzione

Settore/prodotto Settore/prodotto